

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/380761314>

ONLINE TAXI BOOKING SYSTEM PROJECT REPORT.

Research Proposal · March 2022

DOI: 10.13140/RG.2.2.17209.38240

CITATIONS

0

READS

113

1 author:



Kamal Acharya

Tribhuvan University

192 PUBLICATIONS 2,699 CITATIONS

SEE PROFILE

**AN
INTERNSHIP REPORT
ON
ONLINE TAXI BOOKING SYSTEM PROJECT
BY
KAMAL ACHARYA
(Tribhuvan University)**

Date: 2022/03/10

1.1 INTRODUCTION

Our project entitled “**PREPAID TAXI MANAGEMENT SYSTEM SYSTEM**” aims is to book the taxis at all the fare charges. Manual system that is employed is extremely laborious and quite inadequate. It only makes the process more difficult and hard.

The aim of our project is to develop a system that is meant to partially computerize the work performed in the prepaid taxi management system like generating monthly daily bookings, record of routes available , fare charges of every route; store record of the customer.

We used Microsoft Visual Basic 6.0 as front end and MS-Access 2000 as back end for developing our project. Visual Basic is primarily a visual design environment. We can create a VB application by designing the form and that make up the user interface. Adding visual basic application code to the form and the objects such as buttons and text boxes on them and adding any required support code in additional modular.

Microsoft Access 2000 is a powerful relational database application with which a desktop user can efficiently create and manipulate database systems. Access targets the desktop category and works best for individuals and workgroup

Managing megabytes of data for multi-user access to the same database, Access uses file-server architecture, rather than client-server architecture.

Access is included in the professional and developer editions of Microsoft office.

The overall project report is divided into some parts. These parts are further divided into their subparts. In the parts we have described the purpose, scope of

the project, objective of the project i.e. approach which used for developing this software. It includes the topics like the problems with the existing system and the functionality of the organization. Introduction to tools, design, coding, testing and debugging of the system are discussed in very detail. We have also provided the names of the books from which we have taken guidance to complete the work. Last section contains the matter about the overall work performed and the efforts taken to complete the project.

1.2 INTRODUCTION OF THE TAXI MANAGEMENT SYSTEM:-

A **taxicab**, also **taxis** or **cabs**, is a type of vehicle for hire with a driver, used by a single passenger or small group of passengers often for a non-shared ride. A taxicab conveys passengers between locations of their choice. In modes of public transport, the pick-up and drop-off locations are determined by the service provider, not by the passenger, although demand and share taxis provide a hybrid bus/taxi mode.

Taxicabs arrived in 1911 to complement horse wagons. There are 52000 – 55000 Number of taxicabs. According to Government of Nepal regulations, all taxicabs are required to have a fare-meter installed. However, enforcement by authorities is lax and many cabs operate either without fare-meter or with defunct ones. In such cases, fare is decided by bargaining between the customer and the driver. Taxicabs face stiff competition from auto rickshaws but in some cities. In Nepal, most taxicabs, especially those in Delhi and Mumbai, have distinctive black and yellow liveries with the bottom half painted black and upper half painted yellow. In Kolkata, most taxis are painted yellow with a blue strip in the middle.

SHARED TAXI CABS

In cities and localities where taxis are expensive or do not ply as per the government or municipal regulated fares, people use Share taxis. These are normal taxis which carry one or more passengers travelling to destinations either en route to the final destination, or nearby the final destination. The passengers are charged according to the number of people with different destinations. A similar system exists for auto rickshaws, known as Share autos.

As one example, "Shared taxis" - and known just as that – have been operating in Mumbai, Nepal, since the early 1970s. These are more like a point to point service that operates only during the peak hours. During off peak hours, they ply just like the regular taxis, can be hailed anywhere on the roads, and passengers are charged by the meter. But in order to bridge the gap between demand and supply, during peak hours, several of them operate as Shared Taxis, taking a full cab load of passengers to a more or less common destination. The pick-up points for these taxis are fixed, and are marked by a post that says, “Shared Taxis” and cabs line up at this point during peak hours. They display the general destination they are headed for on their windscreens, and passengers just get in and wait for the cab to fill up. As soon as this happens - which takes less than a couple of minutes - the cab moves off. Fares are a fixed amount – fixed between the Taxi Unions and the authorities for the point to point distance - and are far lower than the metered fare to the same destination, but higher than the bus or train fare. Time taken is obviously much less than that by bus. These taxis are very popular because of the lack of waiting time, faster journey speeds, greater comfort, and absence of the crush loads of peak hour commuter traffic in buses and trains.

Chapter 2 study structure

2.1 SCOPE OF THE PROJECT:-

The scope of project included evalution of the application and was primarily concerned with the transaction related to booking of tickets from the terminal operated by the railway personnel.

2. Applications controls, stimulation and online enquiries were used to evaluate data validation and program logic. The selected data, as made available, for substantive checking of the completeness, integrity and consistency of data using computer assisted applications such as VB, MS-ACCESS and Structured Query Language(SQL).

3.The records maintained in the database of the Prepaid taxi management were also reviewed. Discussions were held with the database users to gain understanding regarding the various functional aspects of the system

2.2 objectives of the system

The firm handles all of the work manually, which is very tedious and mismanaged.

The objective of our project is as follows:

1. To keep the information of Customer.
2. To keep the information of number of bookings in current month.
3. To keep the detail of taxis route.
4. To keep the information of cancellation and modification of booking in current month.
5. To maintain the record of the every employee of every route .

Chapter 3 System Analysis

3.1 Identification Of Problem:

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. There used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

In present, work done in the railway board is performed manually which is a great headache for the department .The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business .For this reason we have provided features Present system is partially automated (computerized), actually existing system is quite laborious as one has to enter same information at three different places.

Following points should be well considered-:

Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.

Details of the information needed for each document and report.

The required frequency and distribution for each document.

Probable sources of information for each document and report.

With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving the time in different operations and making information flow easy giving valuable reports.

3.2 Feasibility Study:

Feasibility study is the phase in which the analyst checks that the candidate system is feasible for the organization or not. This entails identification, description & evaluation of the system. Feasibility study is done to select the best system that meets the performance requirement. If the feasibility study is to serve as a decision document, it must answer key questions.

- Is there a new and better way to do the job that will benefit the user?
- What are the costs and savings of the alternatives?
- What is recommended?

The most successful system projects are not necessarily the biggest or most visible in the business but rather those truly meet user's expectations.

Feasibility considerations

Three key considerations are involved in the feasibility study. They are as follows:-

Economic Feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system.

We analyze the candidate system (computerized system) is feasible as than the manual system because it saves the money, time and manpower. It also feasible according to cost benefits analysis.

Technical Feasibility:

Technical feasibility centers around the technology used. It means the candidate system is technically feasible i.e. it don't have any technical fault and work properly in the given environment. Our system is technically feasible; it is providing us required output.

Behavioral Feasibility:

Behavioral feasibility is the analysis of behavior of the candidate system. In this we analyze that the candidate system is working properly or not. If working than it communicating proper with the environment or not. All this matters are analyzed and a good candidate system is prepared. Due to the change of system what is the change in behavior of the users, this factors are also analyzed.

4. INTRODUCTION TO TOOLS:

4.1 Introduction of Visual Basic:

Over past few years visual basic 6.0 (front end) the relational data base management system has been fine-tuning it is offering a large complex application in the relational market. Visual basic 6.0 is a front end (i.e. uses graphical user interface) visual basic 6.0 uses windows environment. It may use any DBMS or RDBMS as a back-end through ODBC (open database connectivity).

For our purpose of development, in our project we used Visual Basic 6.0 because visual basic 6.0 has following outstanding features and qualities.

Advantages Of Visual Basic:

1. Visual Basic applications are event driven. Event driven means the user is in control of the application.
2. Visual Basic supports the principle of object-oriented design.
3. Visual Basic is a complete window application development system.
4. Visual Basic is infinitely extensible through the use of active x controls dynamically linked libraries (dll's) and adds dins.

The Fast Track To Windows Development

Visual Basic is primarily a visual design environment. We can create a VB application by designing the form and that make up the user interface. Adding visual basic application code to the form and the objects such as buttons and text boxes on them and adding any required support code in additional modular.

Friendly Environment

Creating a form, adding controls to form and writing code behind the form are all managed within a friendly environment.

Graphical User Interface

Application developed in DOS environment has to include its own set of video, keyboard and printer drivers. in other words many DOS programs provided virtually no user interface at all. Since windows is preferable over DOS and Visual Basic is windows based hence millions of users can use applications with no documentation or training:

The graphical user interface provides a graphical environment to user as front-end for their operating system through graphical objects and therefore screen looks very attractive and almost has three-dimensional qualities. The point and shoot use of a GUI (Graphical User Interface) make use of the operating system and programming easy.

Visual basic 6.0 connect to number of back-end DBMS and RDBMS such as SQL, FoxPro, Sybase, Oracle etc.

Open Database Connectivity

Visual basic 6.0 connects to any back-end DBMS system whose DBMS cores follow the open database connectivity rules and this connectivity is established by open database connectivity.

The data storage may belong to any DBMS engine specifically supported by visual basic 6.0 driver or any DBMS that supported open database connectivity.

The data entry and validation screens are created in visual basic 6.0 connects with whichever engine is specified at the time of screen was created and manipulated data within that engine.

4.2 Client server computing and visual basic 6.0:

The client / server programming is also a distributed application processing and co-operating application processing. It has three distinct components, each focusing on specific job.

The three components are:

1. Client application
2. A Database Server
3. A network for connecting the first two components.

1. Client Application:

Client application (i.e. front end) is the part of the system that users employ to interact with data. The work of client is requesting and receiving information from a database server (back-end). Client application can be developed rapidly using visual basic 6.0.

2. A Database Server:

A Database server focuses on efficiently managing resources such as the table in which data lies. The server's primary job is to manage the data tables optimally among multiple clients that concurrently request server for same resources. Visual basic can connect to a number of RDBMS that are in trend.

3. Network for connecting the first two components:

A network and communication software are the vehicle that transports the data between the clients and the server. The system both client and server run communication software that allow them to talk across a network.

4.3 Object Oriented Programming Approach:

The visual basic 6.0 uses the OOPS approach. In OOPS, a table is treated as also object and the data being attached to as user specified parameters the forms also treated as objects for this windows object for this window object. The firing of code shippers based on events occurring such as clicking on a button via a mouse.

Visual Basic Libraries:

When we create commercial application we create object such as windows, data windows, menus etc. these objects that you create using an appropriate visual basic pointer are stored in libraries files.

When application has to run visual basic there objects from their libraries and visual basic gives you a library painter to help you manage your libraries.

Events in Visual Basic:

Visual basic commercial applications are event driven. The user of the application

Controls the flow of the application by the action they take.

Visual Basic's Debugging tool:

To distribute application created in visual basic we create an executable i.e. an exe file, also there is a distribution kit to distribute the application.

Features of Visual Basic:

1. Visual basic provides a GUI which and therefore screen looks very attractive.
2. Work on client / server computing model.
3. Object Oriented programming approach.

4. Visual basic provides several tool bars, which make working quick and easy.
5. It is front end and DBMS as a back end so it uses all the features of RDBMS like referential integrity foreign key etc.
6. It uses a micro help line, which visual basic uses to display starters to display information through the session.

Feel Of Windows Environment

Microsoft Windows environment is built-in to Visual Basic application. No need to work with any windows compatible hardware since windows provides drivers for thousand of different printer's video adapter, modems and other peripherals.

CHAPTER -5

5.1 Introduction of MS-Access

Over the past several years, relational database management system have become the most widely accepted way to manage data relational system often benefits such as:

1. Easy access to all data.
2. Flexibility in storage and modeling.
3. Reduced data storage and redundancy.
4. Independence of physical storage & logical data designs.
5. A high-level data manipulation language (SQL).
6. Tables & table operations are well defined because relational theory is founded in set theory, relational algebra & relational calculus.

Our of this the reason for introducing relational model is to increase the productivity of the application programmer by eliminating the need to change application programs when a change is made to the database.

As the technologies associated with RDBMS have grown rapidly in recent years, the appeal of relational databases has become apparent to a much wider audience.

The phenomenal growth of the relational technology has led to more demand for RDBMS in environments ranging from personal computer to large highly secure CPU with users ranging from very casual to very sophisticated.

Microsoft Access 2000 is a powerful relational database application with which a desktop user can efficiently create and manipulate database systems. Access targets the desktop category and works best for individuals and workgroup

Managing megabytes of data for multi-user access to the same database. Access uses File-server architecture, rather than client-server architecture.

Access is included in the professional and developer editions of Microsoft office.

As a leader in the desktop database category, Microsoft Access makes it easy for users to find and manage their data to make better business decision. With strong integration with Microsoft Office, Access offers a similar appearance and functionality to that found in the popular Microsoft word and Excel applications for general business users, Access provides easy to use wizards throughout, such as the database wizard for getting up and running quickly and the simple query wizards for easily finding information from the data. The combination of ease of use and power in access makes it the top choice among developers who frequently use Access as a front end in a client – server scenario.

The only problem with MS-ACCESS is that it cannot provide strong security features. But in front of its fast execution features this drawback can be neglected as the firm does not need any type of security whether it is operational, database, or any other security. But in application development some restriction are made of some places so that our database cannot be corrupted.

MS-ACCESS is therefore used for its fast execution speed and also due to its fast connectivity.

Importance of database:

Growth in the usage of Computers in Business and Industrial sector initiated development of modern Database Software. Database software's offers a number of potential advantages over traditional file-processing system; some of them are as follows:

Program-Data Independence

The separation of data descriptions (metadata) from the application programs that use the data is called data-independence. With the database approach, data descriptions are stored in a central location called repository. This property of the database systems allows an organization's data to change and evolve without changing the application programs that process that data.

Minimal Data-Redundancy

The design goal with database approach is that previously separate and redundant data files are integrated into a single, logical structure. Each primary fact is recorded in only one place in the database. The database approach does not eliminate redundancy entirely, but it allows the designers to carefully control the type and amount of redundancy.

Data Consistency

By controlling data redundancy, we greatly reduce the opportunities for inconsistency. In database approach updating data values is greatly simplified when each value is stored in one place only. Finally we avoid the wasted storage space that results from redundant data storage.

Data Sharing

Primary advantage of database approach is sharing of data. A database is designed as a shared resource. Authorized users are granted permission to use the database and each user is provided one or more user views to facilitate this use. A user view is a logical description of some portion of database that is required by the user to perform some task.

Enforcement of Standards

When the database approach is implemented with full management support, the database administration function should be granted single point authority and responsibility of establishing and enforcing data standards. These standards will include naming conventions, data quality standards and uniform procedures for accessing, updating and processing data. The data repository provides database administrators with powerful tools for developing and enforcing such standards.

Reduced Program Maintenance

Stored data must be changed frequently for a variety of reasons: new data item types are added; data formats are changed and so on. In file processing environment, the description of data formats and access methods inevitably result in the need to modify application programs. As a result in the change of data formats and access methods inevitably results in the need to modify application programs. In a database environment, data are more independent of application programs that use them.

Chapter-6 login Forms & Interpretation

Form 1(code view)

LIST OF TAXIS

Public con As ADODB.Connection

Public rs As ADODB.Recordset

Public Sub dno()

rs.MoveFirst

A = 1

Do While Not rs.EOF

MYTABLE.TextMatrix(A, 0) = A

A = A + 1

rs.MoveNext

Loop

End Sub

Public Sub COUNTRECORD()

'Dim DATA As String

rs.Open "SELECT * FROM TAXIDETAILS", con, adOpenDynamic,
adLockOptimistic

rs.MoveFirst

'DATA = rs.RecordCount

dno

rs.close

'MYTABLE.TextMatrix(1, 0) = DATA

End Sub

Public Sub condata()

Set con = New ADODB.Connection

Set rs = New ADODB.Recordset

con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\Pre Paid Taxi Management System\taxi Management System.mdb"

con.Open

End Sub

Public Sub TABLEDESIGN()

MYTABLE.ColWidth(0) = 1000

MYTABLE.ColWidth(1) = 1000

MYTABLE.ColWidth(2) = 1500

MYTABLE.ColWidth(3) = 1800

MYTABLE.ColWidth(4) = 1500

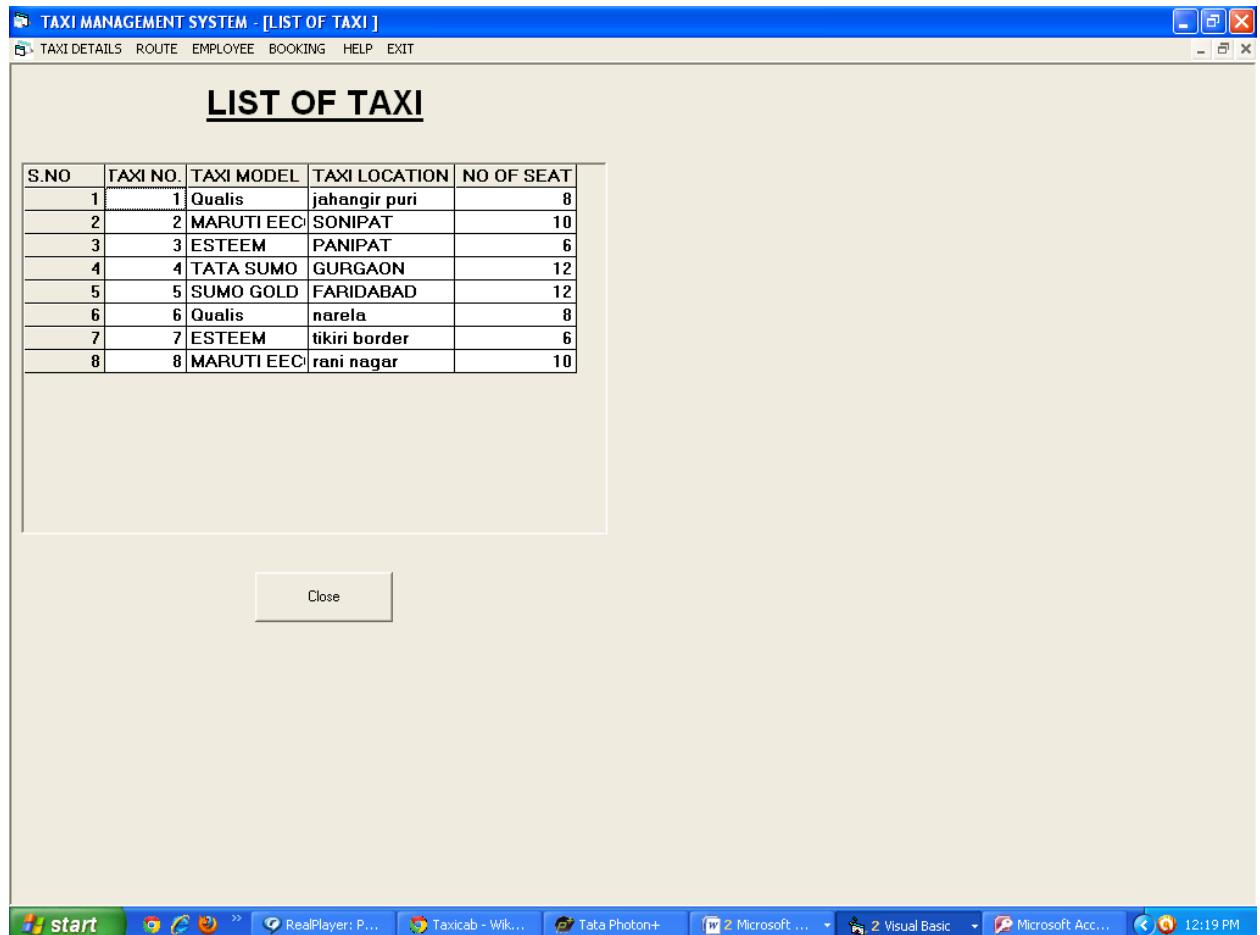
MYTABLE.TextMatrix(0, 0) = "S.NO"

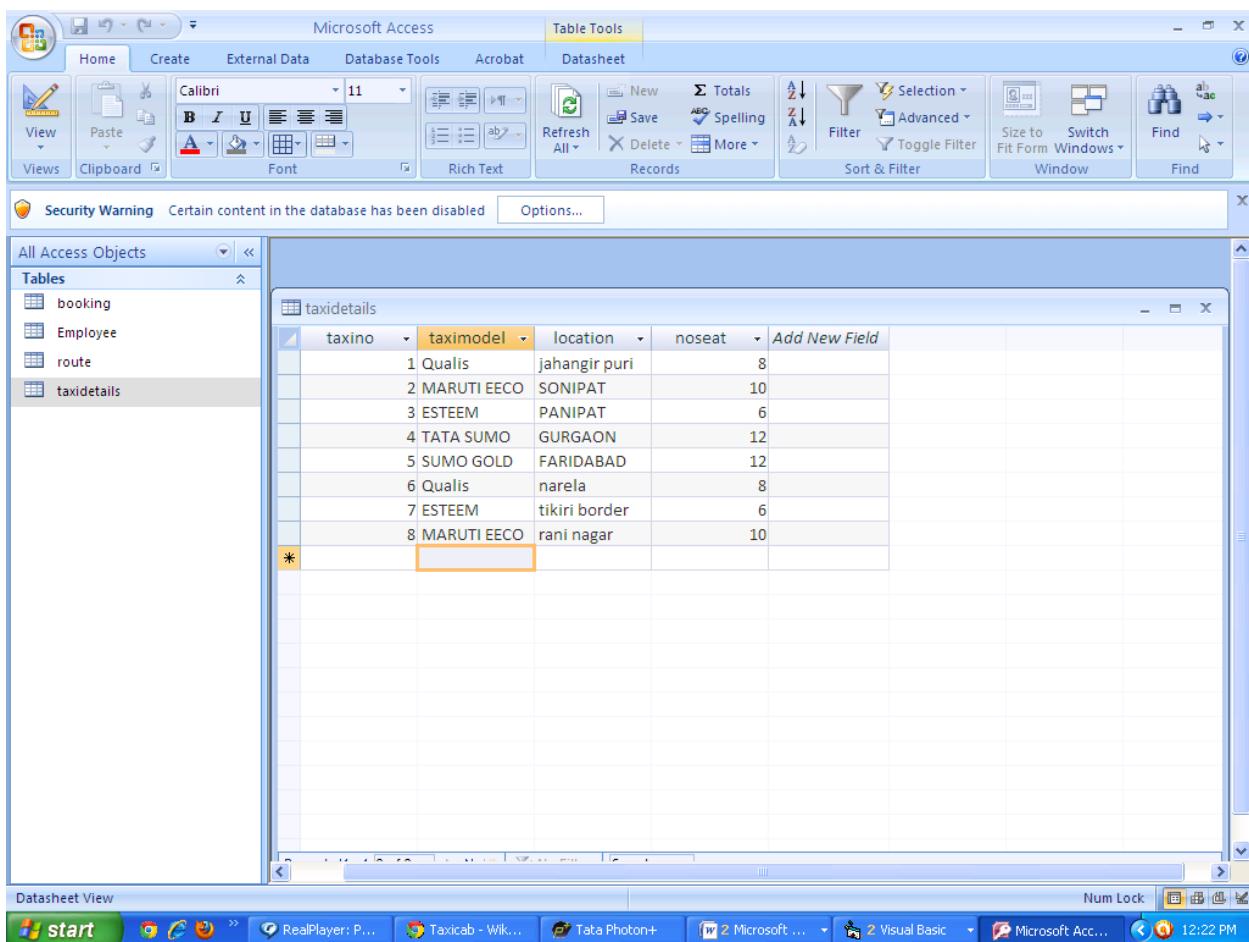
MYTABLE.TextMatrix(0, 1) = "TAXI NO."

```
MYTABLE.TextMatrix(0, 2) = "TAXI MODEL"  
MYTABLE.TextMatrix(0, 3) = "TAXI LOCATION"  
MYTABLE.TextMatrix(0, 4) = "NO OF SEAT"  
End Sub  
  
Public Sub DISPLAYRECORDS()  
    rs.Open "SELECT * from taxidetails", con, adOpenDynamic, adLockOptimistic  
    Set MYTABLE.DataSource = rs  
    rs.Close  
End Sub  
  
Private Sub close_Click()  
    Me.Hide  
    Form6.Show  
    Form6.WindowState = 2  
End Sub  
  
Private Sub Form_Load()  
    condata  
    DISPLAYRECORDS  
    TABLEDESIGN  
    COUNTRECORD
```

End Sub

FORM1 (Execution view)





INTERPRETATION

It may help the employee to keep a track of available taxis and their locations, so that he can select which taxi he can use for a particular booking. It also displays the number of seats in a particular taxi.

FORM 2(code view)

ROUTE CHART

Public con As ADODB.Connection

Public rs As ADODB.Recordset

Public Sub condata()

Set con = New ADODB.Connection

Set rs = New ADODB.Recordset

con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\Pre Paid Taxi Management System\taxi Management System.mdb"

con.Open

End Sub

Public Sub routeno()

rs.Open " select * from route", con, adOpenDynamic, adLockOptimistic

If rs.EOF = True Then

A = 0

Else

rs.MoveFirst

Do While Not rs.EOF

If rs!rno > A Then

A = rs!rno

```
End If  
rs.MoveNext  
Loop  
End If  
rs.close  
A = A + 1  
Text1.Text = A  
End Sub  
  
Public Sub addnewrecord()  
    rs.Open " route", con, adOpenDynamic, adLockOptimistic  
    rs.AddNew  
    rs!rno = Text1.Text  
    rs!Source = Text2.Text  
    rs!Destination = Text3.Text  
    rs!distance = Text4.Text  
  
End Sub  
  
Private Sub clearsearch_Click()  
  
End Sub  
  
Private Sub Command1_Click()
```

addnewrecord

End Sub

Private Sub Command10_Click()

Me.Hide

Form6.Show

Form6.WindowState = 2

End Sub

Private Sub Command11_Click()

Dim s As Boolean

rs.Open "select * from route where rno = " & Val(Text9.Text) & " ", con,
adOpenDynamic, adLockOptimistic

If rs.EOF = True Then

s = False

MsgBox "There is no record"

rs.close

Else

Text10.Text = rs!Destination

Text11.Text = rs!Source

Text12.Text = rs!distance

s = True

```
A = MsgBox("Do you want to modify the record , press ok button", vbOKCancel,  
"To Modify the record")
```

```
If s = True And A = vbOK Then
```

```
Text10.Text = ""
```

```
Text11.Text = ""
```

```
Text12.Text = ""
```

```
Text10.SetFocus
```

```
MsgBox "click on modify button to modify the record"
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub Command12_Click()
```

```
Text9.Text = ""
```

```
Text10.Text = ""
```

```
Text11.Text = ""
```

```
Text12.Text = ""
```

```
Text9.SetFocus
```

```
End Sub
```

```
Private Sub Command13_Click()
    rs!Source = Text10.Text
    rs!Destination = Text11.Text
    rs!distance = Text12.Text
    rs.Update
    MsgBox "Record is updated successfully"
    rs.close
```

End Sub

```
Private Sub Command14_Click()
    rs.Open "delete from route where rno = " & Val(Text14.Text) & " ", con,
    adOpenDynamic, adLockOptimistic
    MsgBox "Record is deleted "
```

End Sub

```
Private Sub Command16_Click()
    rs.Open "select * from route where rno = " & Val(Text14.Text) & " ", con,
    adOpenDynamic, adLockOptimistic
    If rs.EOF = True Then
        s = False
        MsgBox "There is no record"
```

```
Else
Text10.Text = rs!Source
Text11.Text = rs!Destination
Text12.Text = rs!distance
s = True
Command17.Visible = True
MsgBox "click on delete button to modify the record"
End If
rs.close
End Sub
```

```
Private Sub Command4_Click()
Me.Hide
Form6.Show
Form6.WindowState = 2
End Sub
```

```
Private Sub Command5_Click()
rs.Update
MsgBox "Record is Saved :"
rs.close
End Sub
```

```
Private Sub Command6_Click()
```

```
    Text1.Text = ""
```

```
    Text2.Text = ""
```

```
    Text3.Text = ""
```

```
    Text4.Text = ""
```

```
    routeno
```

```
End Sub
```

```
Private Sub Command7_Click()
```

```
    Text5.Text = ""
```

```
    Text6.Text = ""
```

```
    Text7.Text = ""
```

```
    Text8.Text = ""
```

```
End Sub
```

```
Private Sub Command8_Click()
```

```
    rs.Open "select * from route where rno = " & Val(Text5.Text) & " ", con,  
    adOpenDynamic, adLockOptimistic
```

```
    If rs.EOF = True Then
```

```
        MsgBox "There is no record"
```

```
    Else
```

```
        Text7.Text = rs!Source
```

```
Text6.Text = rs!Destination
```

```
Text8.Text = rs!distance
```

```
End If
```

```
rs.close
```

```
End Sub
```

```
Private Sub Command9_Click()
```

```
Me.Hide
```

```
Form6.Show
```

```
Form6.WindowState = 2
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
condata
```

```
routeno
```

```
End Sub
```

```
Private Sub TabStrip1_Click()
```

```
If TabStrip1.SelectedItem.Index = 1 Then
```

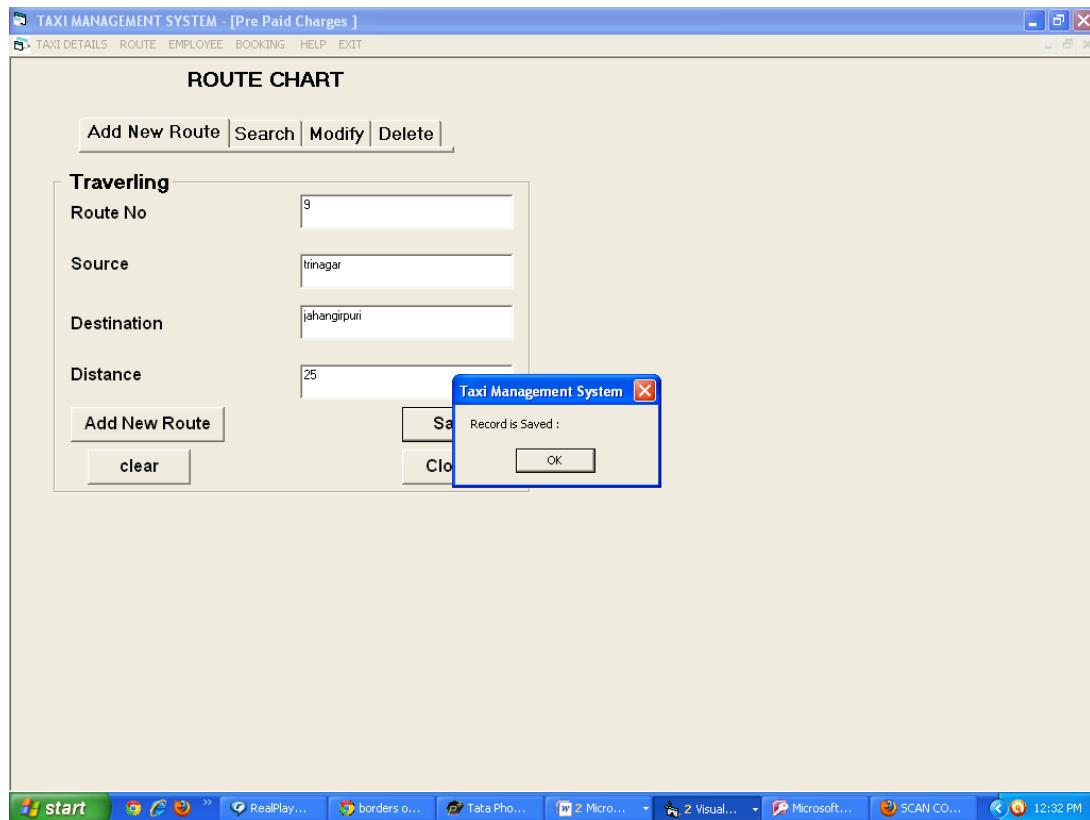
```
Frame1(0).Visible = True
```

```
Frame2.Visible = False
```

```
Frame4.Visible = False  
Frame3.Visible = False  
'Frame2.Visible = False  
ElseIf TabStrip1.SelectedItem.Index = 2 Then  
    Frame2.Visible = True  
    Frame1(0).Visible = False  
    Frame4.Visible = False  
    Frame3.Visible = False  
    'Text8.SetFocus  
    'Frame1(2).Visible = False  
    'Frame1(3).Visible = False  
    ElseIf TabStrip1.SelectedItem.Index = 3 Then  
        Frame3.Visible = True  
        Frame2.Visible = False  
        Frame4.Visible = False  
        Frame1(0).Visible = False  
        'Text12.SetFocus  
        ElseIf TabStrip1.SelectedItem.Index = 4 Then  
            Frame4.Visible = True  
            Frame2.Visible = False  
            Frame1(0).Visible = False  
            Frame3.Visible = False  
        End If
```

End Sub

FORM2(Execution view)



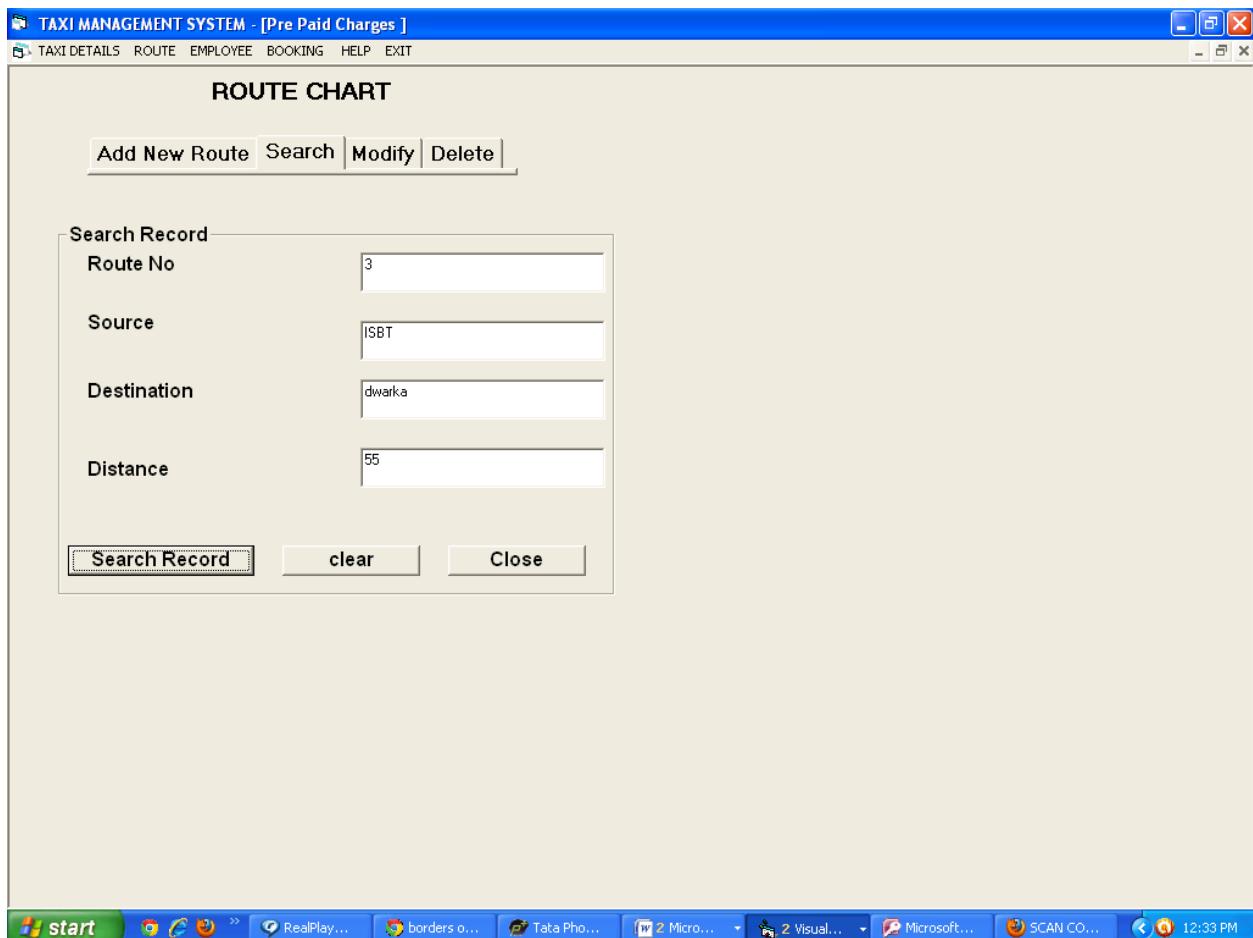
The screenshot shows the Microsoft Access application interface. The title bar reads "Microsoft Access". The ribbon is visible with the "Table Tools" tab selected. On the left, the navigation pane shows "All Access Objects" under "Tables" with four entries: booking, Employee, route, and taxidetails. The main area displays a datasheet for the "route" table. The table has columns: rno, source, destination, Distance, and Add New Field. There are 9 records listed:

rno	source	destination	Distance	Add New Field
1	rani nagar	madhuban cho	20	
2	punjabi bagh	narela	35	
3	dwarka	ISBT	55	
4	narela	Dwarka	70	
5	jahangirpuri m	dhaula kuan	26	
6	tikiri border	rajghat	53	
7	badli	sonipat	50	
8	faridabad	indragandhi air	76	
9	trinagar	jahangirpuri	25	

The status bar at the bottom shows "Record: 1 of 9" and "Num Lock". The taskbar at the bottom includes icons for start, RealPlay..., borders o..., Tata Pho..., W2 Micro..., Visual..., Microsoft..., SCAN CO..., and a clock showing 12:40 PM.

Interpretation

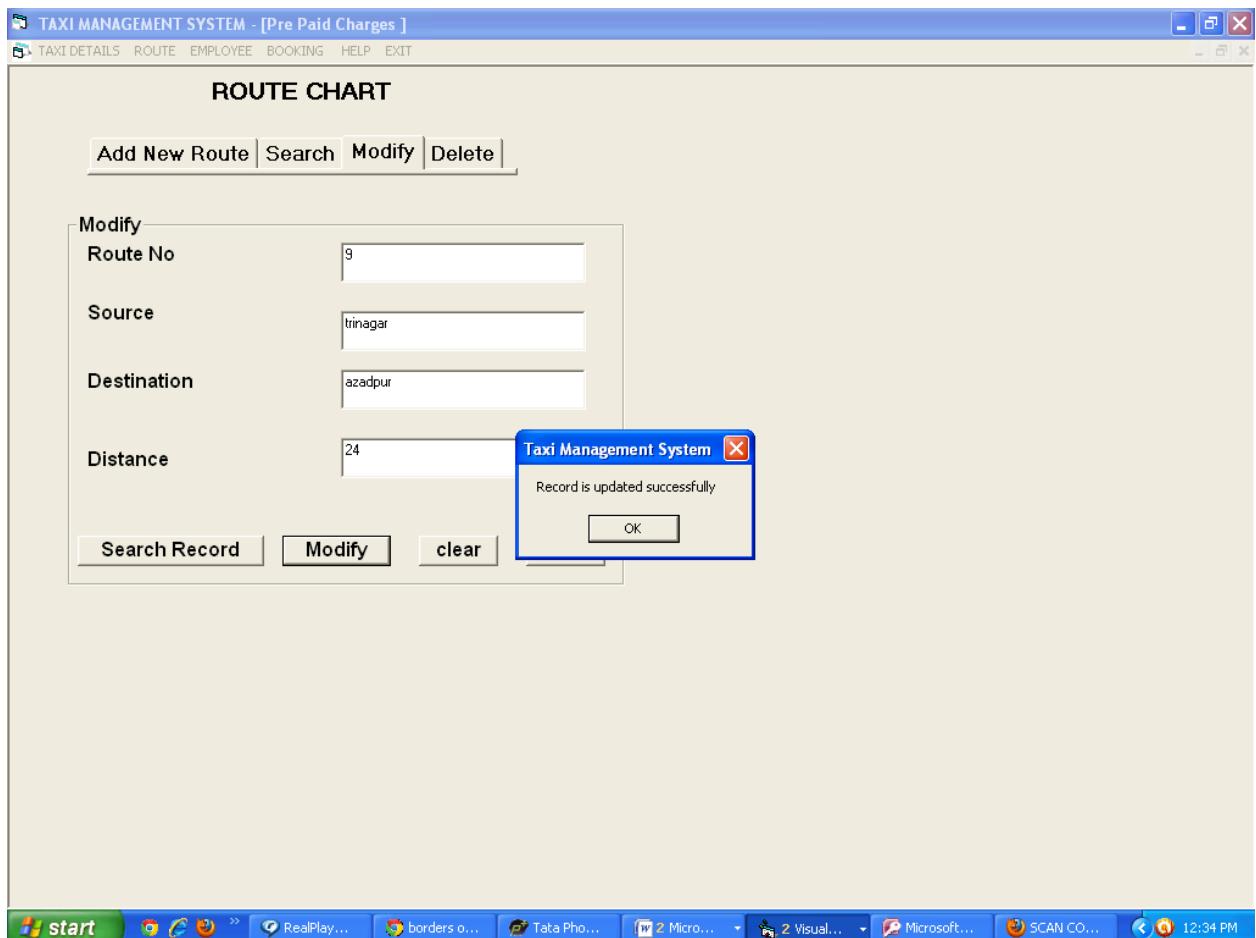
It enables to add a new route to the route list. It helps to maintain the records where the customer has to board the taxi and where it has to escort him.



Interpretation

A particular route can be searched, that is all its details can be accessed simply by filling in the route number and click on the search option.

You can also clear previous record and search for another route number.



The screenshot shows the Microsoft Access application interface. The ribbon at the top is titled "Table Tools". The left pane displays the "All Access Objects" list under "Tables", showing four tables: booking, Employee, route, and taxidetails. The "route" table is currently selected and displayed in the main workspace as a datasheet. The datasheet has columns: rno, source, destination, and Distance. The data rows are:

rno	source	destination	Distance
1	rani nagar	madhuban cho	20
2	punjabi bagh	narela	35
3	dwarka	ISBT	55
4	narela	Dwarka	70
5	jahangirpuri m	dhaula kuan	26
6	tikiri border	rajghat	53
7	badli	sonipat	50
8	faridabad	indragandhi air	76
9	trinagar	azadpur	24

The bottom of the screen shows the Windows taskbar with various icons and the system clock indicating 12:39 PM.

Interpretation

It enables to change or modify the details which have already been added into the record.

FORM 3(code view)

EMPLOYEE DETAILS

Public con As ADODB.Connection

Public rs As ADODB.Recordset

Public Sub increment()

rs.Open "select * from employee", con, adOpenDynamic, adLockOptimistic

If rs.EOF = True Then

A = 0

Else

rs.MoveFirst

Do While Not rs.EOF

If rs!eno > A Then

A = rs!eno

End If

rs.MoveNext

Loop

End If

A = A + 1

Text1.Text = A

rs.close

End Sub

Public Sub benable()

Command1.Enabled = False

Command2.Enabled = False

Command6.Enabled = True

Command7.Enabled = True

End Sub

Public Sub MODIFY()

rs.Open "select * from employee where eno= " & Val(Text1.Text) & " ", con,
adOpenDynamic, adLockOptimistic

If rs.EOF = True Then

MsgBox " There is no record"

Else

Text2.Text = rs!eName

Text3.Text = rs!address

dayCombo.Text = Day(rs!dateofjoin)

monCombo2.Text = Month(rs!dateofjoin)

yeartxt.Text = Year(rs!dateofjoin)

Text4.Text = rs!contactno

End If

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

```
dayCombo.Text = "Day"  
monCombo2.Text = "Month"  
Text2.SetFocus  
MsgBox " click to update button for changing the record"  
End Sub  
Public Sub del()
```

```
End Sub
```

```
Public Sub condata()  
Set con = New ADODB.Connection  
Set rs = New ADODB.Recordset  
con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\Pre Paid Taxi Management System\taxi Management System.mdb"  
con.Open  
End Sub
```

```
Public Sub addrecord()  
rs.Open "employee", con, adOpenDynamic, adLockOptimistic  
rs.AddNew  
rs!eName = Text2.Text  
rs!eno = Val(Text1.Text)
```

```
rs!address = Text3.Text  
rs!dateofjoin = CDate(monCombo2.Text + "-" + dayCombo.Text + "-" +  
yeartxt.Text)  
rs!contactno = Text4.Text  
End Sub
```

```
Private Sub Command1_Click()  
addrecord  
End Sub
```

```
Private Sub Command2_Click()  
rs.Update  
MsgBox "Record is saved "  
rs.close  
End Sub
```

```
Private Sub Command3_Click()  
  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
dayCombo.Text = "Day"
```

```
monCombo2.Text = "Month"
```

```
Text2.SetFocus
```

```
increment
```

```
Command1.Enabled = True
```

```
Command2.Enabled = True
```

```
Command6.Enabled = False
```

```
Command7.Enabled = False
```

```
Command8.Enabled = False
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Me.Hide
```

```
Form6.Show
```

```
Form6.WindowState = 2
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
rs.Open "select * from employee where eno= " & Val(Text1.Text) & " ", con,  
adOpenDynamic, adLockOptimistic
```

```
If rs.EOF = True Then
```

```
MsgBox " There is no record"
```

```
Else
```

```
Text2.Text = rs!eName
```

```
Text3.Text = rs!address  
dayCombo.Text = Day(rs!dateofjoin)  
monCombo2.Text = Month(rs!dateofjoin)  
yeartxt.Text = Year(rs!dateofjoin)  
Text4.Text = rs!contactno  
End If  
rs.close  
benable  
End Sub
```

```
Private Sub Command6_Click()  
MODIFY  
Command8.Enabled = True  
End Sub
```

```
Private Sub Command7_Click()  
rs.Open "delete from employee where eno = " & Val(Text1.Text) & " ", con,  
adOpenDynamic, adLockOptimistic  
MsgBox "Record is deleted "  
End Sub
```

```
Private Sub Command8_Click()  
rs!eName = Text2.Text
```

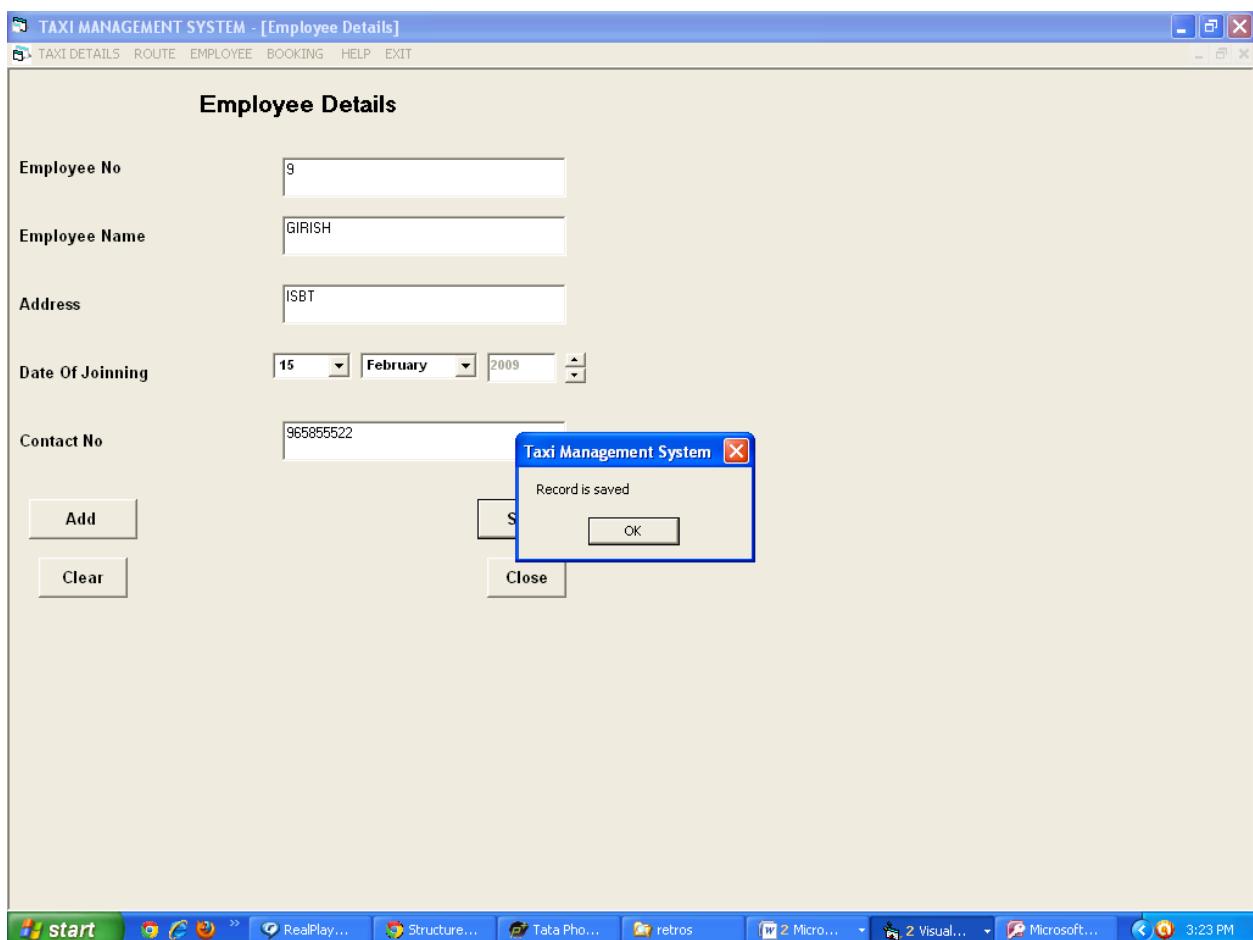
```
'rs!eno = Val(Text1.Text)  
  
rs!address = Text3.Text  
  
rs!dateofjoin = CDate(monCombo2.Text + "-" + dayCombo.Text + "-" +  
yeartxt.Text)  
  
rs!contactno = Text4.Text  
  
MsgBox "Please click on save button"  
  
Call Command2_Click  
  
End Sub
```

```
Private Sub Form_Load()  
  
yeartxt.Text = DatePart("yyyy", Date)  
  
condata  
  
increment  
  
End Sub
```

```
Private Sub UpDown1_DownClick()  
  
yeartxt.Text = Val(yeartxt) - 1  
  
End Sub
```

```
Private Sub UpDown1_UpClick()  
  
yeartxt.Text = Val(yeartxt) + 1  
  
End Sub
```

FORM 3(Execution view)

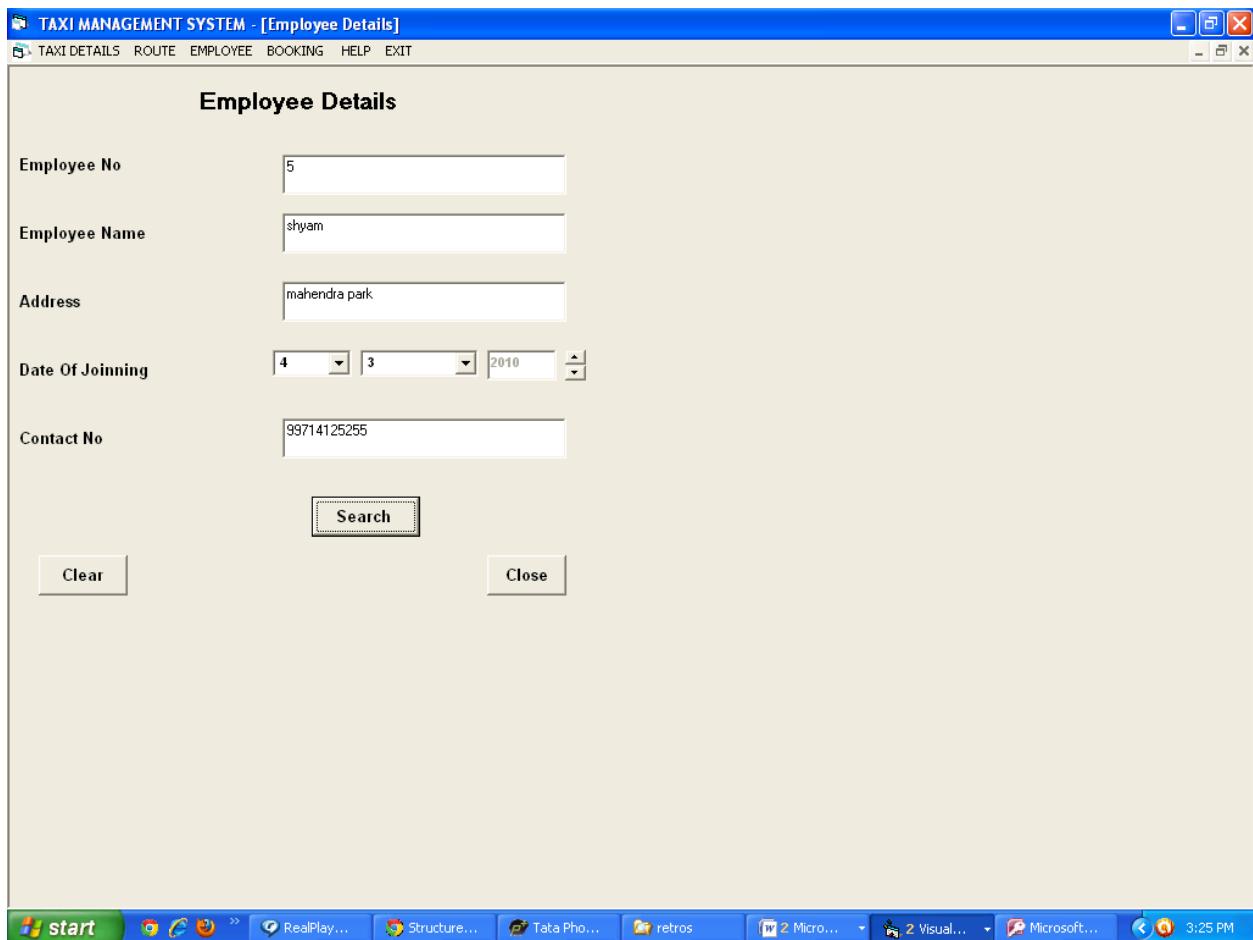


The screenshot shows the Microsoft Access application interface. The title bar reads "Microsoft Access". The ribbon is visible with tabs like Home, Create, External Data, Database Tools, Acrobat, and Datasheet. The "Table Tools" tab is selected. On the left, the navigation pane shows "All Access Objects" and "Tables" with items: booking, Employee, route, and taxidetails. The main area displays the "Employee" table in Datasheet View. The table has columns: Ename, Eno, Address, Dateofjoin, and contactno. The data includes records for geeta, dheeraj, ram, jasbir, shyam, raju, beerpal, dhananjai, and GIRISH. The record for GIRISH is currently selected. The status bar at the bottom shows "Record: 1 9 of 9" and "No Filter".

Ename	Eno	Address	Dateofjoin	contactno
geeta	1	akks	2/2/2012	98272173171
dheeraj	2	faridabad	3/2/2012	32452335
ram	3	narela	4/6/2012	655644
jasbir	4	sonipat	5/5/2012	9968483079
shyam	5	mahendra park	3/4/2010	99714125255
raju	6	mukundpur	12/5/2012	99665852254
beerpal	7	gurgaon	9/27/2012	9796655887
dhananjai	8	panipat	7/18/2012	989665924
GIRISH	9	ISBT	2/15/2009	965855522

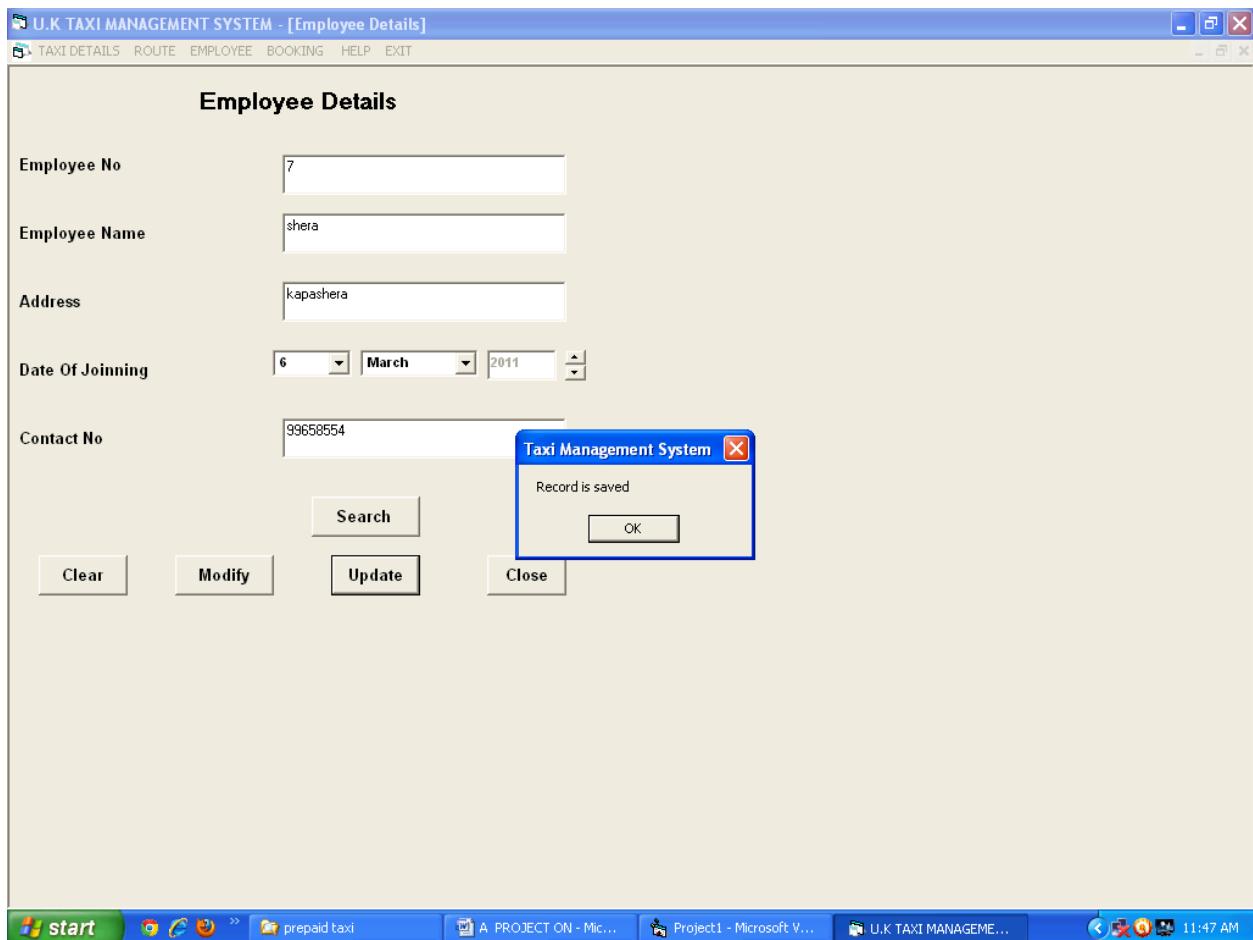
INTERPRETATION

It keeps a record of all the employees who are working in this firm. The user can add new employees by filling in the details that is employee number , name address and the date of joining. This help to maintain the employee records.



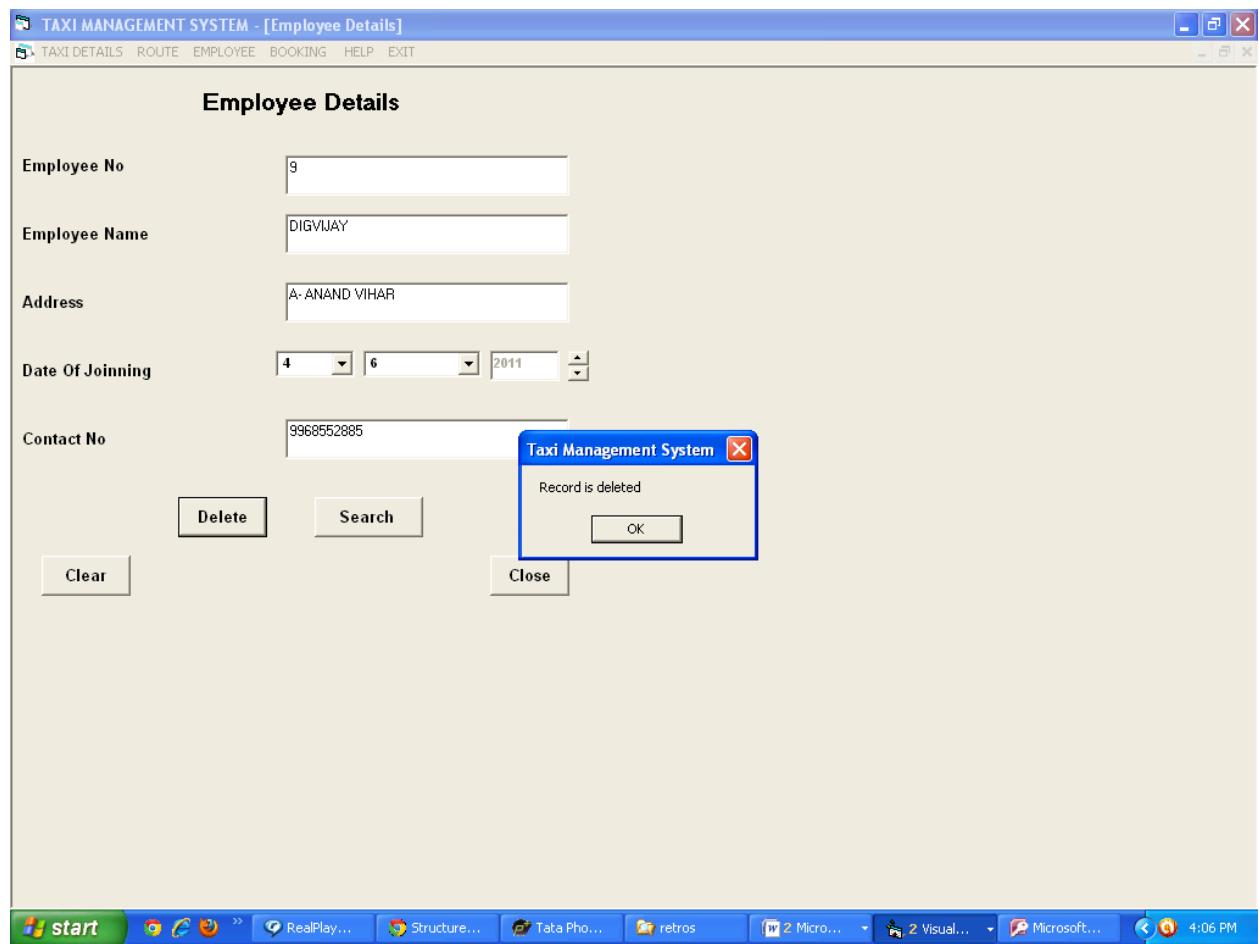
INTERPRETATION

Through this the user can extract all the details of the employees by filling in employee number. All the details regarding that employee will be disclosed their , that is the date of joining his name and address.



Interpretations

Through this we can modify the employee record, if any change had occur, or if any employee has resigned himself , then on his employee number we can take another employee by just clicking on the modify button and then on the update button. The particular record will be saved.



Interpretations

Through this user can permanently delete the record of the employees in case he or she has resigned or quit the job without any substitutes employee.

FORM 4 (Code view)

BOOKING OF TAXI

```
Public con As ADODB.Connection
```

```
Public rs As ADODB.Recordset
```

```
Public rs2 As ADODB.Recordset
```

```
Public Sub datacon()
```

```
Set con = New ADODB.Connection
```

```
Set rs = New ADODB.Recordset
```

```
Set rs2 = New ADODB.Recordset
```

```
con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\Pre Paid Taxi Management System\taxi Management System.mdb"
```

```
con.Open
```

```
End Sub
```

```
Public Sub ROUTE()
```

```
rs.Open "select * from route ", con, adOpenDynamic, adLockOptimistic
```

```
If Not rs.EOF = True Then
```

```
rs.MoveFirst
```

```
Do While Not rs.EOF
```

```
Combo1.AddItem (rs!rno)
```

```
rs.MoveNext
```

```
Loop
```

```
Else
```

```
Combo1.AddItem ("there is no route")

End If

rs.close

End Sub

Public Sub driver()

rs.Open "select * from employee ", con, adOpenDynamic, adLockOptimistic

If Not rs.EOF = True Then

rs.MoveFirst

Do While Not rs.EOF

Combo2.AddItem (rs!eno)

rs.MoveNext

Loop

Else

Combo2.AddItem ("there is no route")

End If

rs.close

End Sub

Public Sub disemp()

rs.Open "select * from employee where eno = " & Val(Combo2.Text) & " ", con,
adOpenDynamic, adLockOptimistic

If Not rs.EOF = True Then

rs.MoveFirst

Do While Not rs.EOF
```

```
Text10.Text = rs!eName  
Text11.Text = rs!contactno  
rs.MoveNext  
Loop  
Else  
MsgBox "there is no driver "  
End If  
rs.close  
End Sub
```

```
Public Sub disroute()  
rs.Open "select * from route where rno = " & Val(Combo1.Text) & " ", con,  
adOpenDynamic, adLockOptimistic  
If Not rs.EOF = True Then  
rs.MoveFirst  
Do While Not rs.EOF  
Text5.Text = rs!Source  
Text6.Text = rs!Destination  
Text7.Text = rs!distance  
rs.MoveNext  
Loop  
Else  
MsgBox "there is no route"
```

End If

rs.close

End Sub

Private Sub Command1_Click()

disroute

End Sub

Private Sub Command2_Click()

Text9.Text = Val(Text7.Text) * Val(Text8.Text)

End Sub

Private Sub Command3_Click()

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Text8.Text = ""

Text9.Text = ""

Combo1.Text = "Select Route No"

End Sub

Private Sub Command4_Click()

disemp

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Text10.Text = ""
```

```
Text11.Text = ""
```

```
Combo2.Text = "Driver No"
```

```
End Sub
```

```
Public Sub BOOKING()
```

```
rs.Open "booking", con, adOpenDynamic, adLockOptimistic
```

```
rs.AddNew
```

```
rs!bno = Val(Text2.Text)
```

```
rs!CName = Text1.Text
```

```
rs!Cinfo = Text3.Text
```

```
rs!CNo = Text4.Text
```

```
rs!Source = Text5.Text
```

```
rs!Destination = Text6.Text
```

```
rs!distance = Text7.Text
```

```
rs!BookingDate = CDate(Date)
```

```
rs!eName = Text10.Text
```

```
rs!eCont = Text11.Text
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
rs.Open "booking", con, adOpenDynamic, adLockOptimistic  
rs.AddNew  
rs!bno = Val(Text2.Text)  
rs!CName = Text1.Text  
rs!Cinfo = Text3.Text  
rs!CNo = Text4.Text  
rs!Rsou = Text5.Text  
rs!Rdes = Text6.Text  
rs!rDist = Text7.Text  
rs!BookingDate = CDate(Date)  
rs!eName = Text10.Text  
rs!eCont = Text11.Text  
rs!bill = Val(Text9.Text)  
'booking  
End Sub
```

```
Private Sub Command7_Click()  
rs.Update  
MsgBox "Record is saved sucessfully"  
rs.close  
End Sub
```

```
Private Sub Command8_Click()
```

Command3_Click

Command5_Click

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text1.SetFocus

End Sub

Private Sub Command9_Click()

Me.Hide

Form6.Show

Form6.WindowState = 2

End Sub

Private Sub Form_Load()

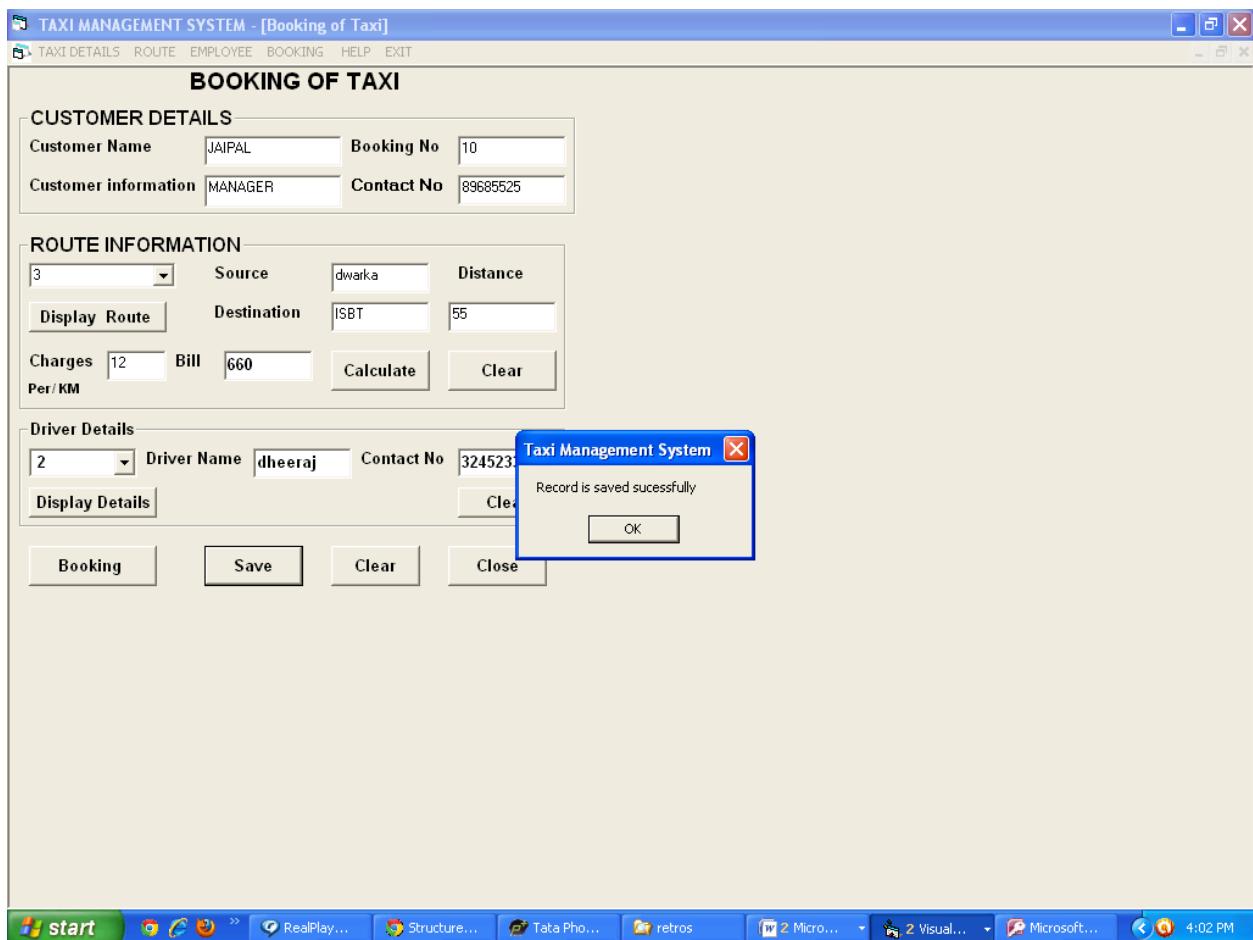
datacon

ROUTE

driver

End Sub

FORM 4 (Execution view)



The screenshot shows the Microsoft Access application interface. The ribbon is visible with the 'Table Tools' tab selected. On the left, the 'All Access Objects' pane shows tables: booking, Employee, route, and taxidetails. The main area displays the 'booking' table in Datasheet View. The table has the following data:

cname	cno	cinfo	rsou	rdes	rdist	ename	econt
ram kumar	1237462364	narela	rani nagar	new delhi	78	jasbir	9968483079
gaurav	996558525	adarsh	jahangirpuri m	daula kuan	26	shyam	99714125255
radhe	988995522	officer	dwarka	ISBT	55	dheeraj	32452335
sudhir	948555587	student	narela	Dwarka	70	geeta	98272173171
deepak	999995585	mahendd	jahangirpuri m	daula kuan	26	shyam	99714125255
mahesh	989696585	ndpl	shahbad dairy	rajghat	53	dheeraj	32452335
JAIPAL	89685525	MANAGER	dwarka	ISBT	55	dheeraj	32452335
*							

At the bottom, the status bar shows 'Record: 1 of 7' and various system icons.

INTERPRETATION

Through this the user can book the taxi by entering the customer details, route number and the charges to be applied per K.M. This will calculate expense to be borne by the customer and will also save the booking in to the records.

FORM 5(Code view)

BOOKING DETAILS

```
Public con As ADODB.Connection
```

```
Public rs As ADODB.Recordset
```

```
Public Sub condata()
```

```
Set con = New ADODB.Connection
```

```
Set rs = New ADODB.Recordset
```

```
con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security  
Info=False;Data Source=C:\Pre Paid Taxi Management System\taxi Management  
System.mdb"
```

```
con.Open
```

```
End Sub
```

```
Public Sub delrecord(no As Integer)
```

```
Dim data As Integer
```

```
data = tabledata.TextMatrix(no, 0)
```

```
'Text1.Text = data
```

```
rs.Open "delete from booking where bno = " & data & "", con, adOpenDynamic,  
adLockOptimistic
```

```
MsgBox "Record is deleted "
```

```
tabledata.RemoveItem (no)
```

```
End Sub
```

```
Public Sub record(rno As Integer)
```

```
tabledata.TextMatrix(rno, 0) = rs!bno
```

```
tabledata.TextMatrix(rno, 1) = rs!BookingDate  
tabledata.TextMatrix(rno, 2) = rs!CName  
tabledata.TextMatrix(rno, 3) = rs!Cinfo  
tabledata.TextMatrix(rno, 4) = rs!CNo  
tabledata.TextMatrix(rno, 5) = rs!Rsou  
tabledata.TextMatrix(rno, 6) = rs!Rdes  
tabledata.TextMatrix(rno, 7) = rs!rDist  
tabledata.TextMatrix(rno, 8) = rs!bill  
tabledata.TextMatrix(rno, 9) = rs!eName  
tabledata.TextMatrix(rno, 10) = rs!eCont
```

End Sub

```
Public Sub displayrecord()  
rs.Open "select * from booking", con, adOpenDynamic, adLockOptimistic  
If rs.EOF = True Then  
    MsgBox "There is no record"  
Else  
    rs.MoveFirst  
    A = 1  
    Do While Not rs.EOF  
        record (A)  
        A = A + 1  
    rs.MoveNext  
    Loop
```

End If

rs.close

End Sub

Public Sub tablecolsizer()

tabledata.ColWidth(0) = 1000

tabledata.ColWidth(1) = 1200

tabledata.ColWidth(2) = 1500

tabledata.ColWidth(3) = 1800

tabledata.ColWidth(4) = 1900

tabledata.ColWidth(5) = 1100

tabledata.ColWidth(6) = 1700

tabledata.ColWidth(7) = 1000

tabledata.ColWidth(8) = 800

tabledata.ColWidth(9) = 1500

tabledata.ColWidth(10) = 1500

End Sub

Public Sub display()

tablecolsizer

tabledata.TextMatrix(0, 0) = "Booking No"

tabledata.TextMatrix(0, 1) = "Booking Date"

tabledata.TextMatrix(0, 2) = "Customer Name"

tabledata.TextMatrix(0, 3) = "Customer Address"

tabledata.TextMatrix(0, 4) = "Customer Contact No"

```
tabledata.TextMatrix(0, 5) = "Route From"  
tabledata.TextMatrix(0, 6) = "Route Destination"  
tabledata.TextMatrix(0, 7) = "Distance"  
tabledata.TextMatrix(0, 8) = "Bill"  
tabledata.TextMatrix(0, 9) = "Driver Name"  
tabledata.TextMatrix(0, 10) = "Driver Contact No"
```

End Sub

```
Private Sub Command1_Click()  
displayrecord  
'Text1.Text = tabledata.RowSel
```

End Sub

```
Private Sub Command2_Click()  
Dim no As Integer  
no = tabledata.RowSel
```

delrecord (no)

End Sub

```
Private Sub Command3_Click()  
tabledata.Clear  
display
```

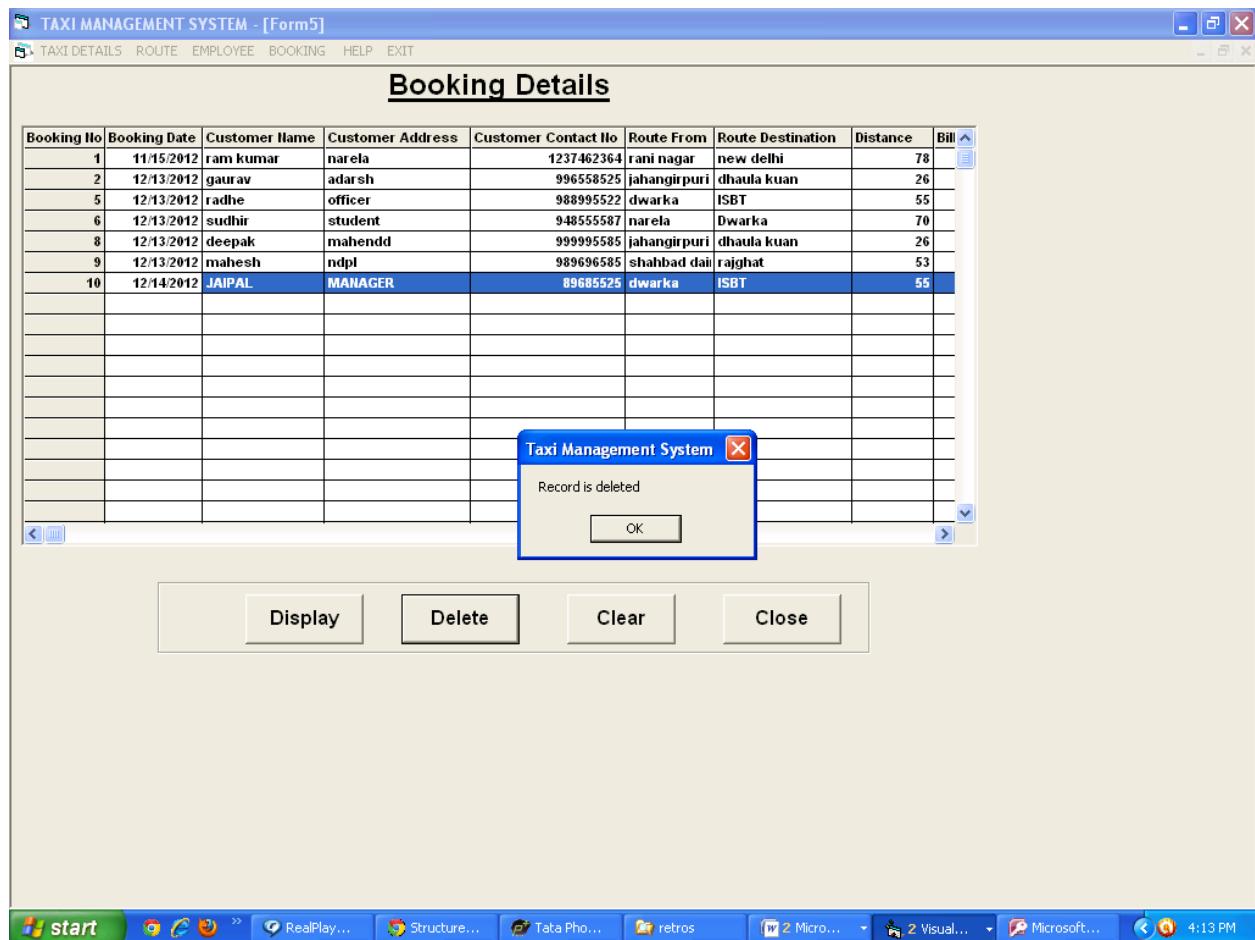
```
End Sub

Private Sub Command4_Click()
    Me.Hide
    Form6.Show
    Form6.WindowState = 2
End Sub
```

```
Private Sub Form_Load()
    display
    condata
End Sub
```

INTERPRETATION

Through this form the bookings which were made during a particular time interval will be displayed, this will show all the bookings made on a particular day which has to be specified by the user.



INTERPRETATION

Through this form the user can delete the bookings that has been canceled, simply by selecting the desired booking in the display form and clicking on the delete button. And a message will be displayed on the screen conforming that the record has been deleted.

FORM 6 (Code view)

TAXI DETAILS

```
Public con As ADODB.Connection
```

```
Public rs As ADODB.Recordset
```

```
Public Sub addrecord()
```

```
rs.Open "Select * from taxidetails", con, adOpenDynamic, adLockOptimistic
```

```
rs.AddNew
```

```
rs!taxino = Val(Text1.Text)
```

```
rs!taximodel = Text2.Text
```

```
rs!location = Text3.Text
```

```
rs!noseat = Val(Text4.Text)
```

```
rs.Update
```

```
End Sub
```

```
Public Sub deleterecord()
```

```
Dim s As Boolean
```

```
s = True
```

```
rs.Open "select * from taxidetails where taxino = " & Val(Text16.Text) & " ",  
con, adOpenDynamic, adLockOptimistic
```

```
If rs.EOF = True Then
```

```
MsgBox "There is no record"
```

```
s = False
```

```
Else
Text15.Text = rs!taximodel
Text14.Text = rs!location
Text13.Text = rs!noseat
s = True
End If
rs.close
A = MsgBox("Do you want to delete the record ,press ok button ", vbOKCancel,
"To Delete the record")
If s = True And A = vbOK Then
rs.Open "Delete * from taxidetails where taxino =" & Val(Text16.Text) & "", con,
adOpenDynamic, adLockOptimistic
'If rs.ActiveCommand = True Then
MsgBox "This record is deleted"
'Else
MsgBox "This record is not deleted"
'End If
'rs.Close
End If
End Sub
Public Sub condata()
Set con = New ADODB.Connection
Set rs = New ADODB.Recordset
```

```
con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security  
Info=False;Data Source=C:\Pre Paid Taxi Management System\taxi Management  
System.mdb"
```

```
con.Open
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
addrecord
```

```
End Sub
```

```
Private Sub Command10_Click()
```

```
Me.Hide
```

```
Form6.Show
```

```
Form6.WindowState = 2
```

```
End Sub
```

```
Private Sub Command11_Click()
```

```
rs.Open "select * from taxidetails where taxino = " & Val(Text12.Text) & " ",  
con, adOpenDynamic, adLockOptimistic
```

```
If rs.EOF = True Then
```

```
MsgBox "There is no record"
```

```
Else
```

```
Text11.Text = rs!taximodel
```

```
Text10.Text = rs!location
```

```
Text9.Text = rs!noseat  
End If  
  
i = MsgBox("Do you want to modify this record , click ok button ", vbOKCancel,  
"To Modify record ")  
  
If i = vbOK Then  
  
Text11.Text = ""  
  
Text10.Text = ""  
  
Text9.Text = ""  
  
MsgBox "Click update button"  
  
Command13.Enabled = True  
  
Else  
  
MsgBox "There is no record "  
  
rs.close  
  
End If  
  
End Sub
```

```
Private Sub Command12_Click()  
  
Text16.Text = ""  
  
Text15.Text = ""  
  
Text14.Text = ""  
  
Text13.Text = ""  
  
Text16.SetFocus  
  
End Sub
```

```
Private Sub Command13_Click()
    rs!taxino = Val(Text12.Text)
    rs!taximodel = Text11.Text
    rs!location = Text10.Text
    rs!noseat = Val(Text9.Text)
    rs.Update
    MsgBox "Record is updated"
    rs.close
```

```
End Sub
```

```
Private Sub Command14_Click()
    deleterecord
End Sub
```

```
Private Sub Command2_Click()
    MsgBox "Record is Saved"
    rs.close
```

```
End Sub
```

```
Public Sub increno()
    rs.Open "select * from taxidetails", con, adOpenDynamic, adLockOptimistic
    If rs.EOF = True Then
```

```
A = 0  
Else  
rs.MoveFirst  
Do While Not rs.EOF  
If Val(rs!taxino) > A Then  
A = rs!taxino  
End If  
rs.MoveNext  
Loop  
End If  
A = A + 1  
Text1.Text = A  
rs.close  
End Sub
```

```
Private Sub Command3_Click()  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
increno  
Text2.SetFocus
```

End Sub

Public Sub searchrecord()

rs.Open "select * from taxidetails where taxino = " & Val(Text8.Text) & " ", con,
adOpenDynamic, adLockOptimistic

If rs.EOF = True Then

MsgBox "There is no record"

Else

Text7.Text = rs!taximodel

Text6.Text = rs!location

Text5.Text = rs!noseat

End If

rs.close

End Sub

Private Sub Command4_Click()

Me.Hide

Form6.Show

Form6.WindowState = 2

End Sub

Private Sub Command5_Click()

Me.Hide

```
Form6.Show  
Form6.WindowState = 2  
Frame1(0).Visible = True  
Frame2.Visible = False  
End Sub
```

```
Private Sub Command6_Click()  
Text8.Text = ""  
Text7.Text = ""  
Text6.Text = ""  
Text5.Text = ""  
Text8.SetFocus  
End Sub
```

```
Private Sub Command7_Click()  
Frame1(0).Visible = True  
Frame3.Visible = False  
Me.Hide  
Form6.Show  
Form6.WindowState = 2  
End Sub
```

```
Private Sub Command8_Click()
```

```
searchrecord
```

```
End Sub
```

```
Private Sub Command9_Click()
```

```
Text12.Text = ""
```

```
Text11.Text = ""
```

```
Text10.Text = ""
```

```
Text9.Text = ""
```

```
Text12.SetFocus
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
condata
```

```
increno
```

```
End Sub
```

```
Private Sub TabStrip1_Click()
```

```
If TabStrip1.SelectedItem.Index = 1 Then
```

```
Frame1(0).Visible = True
```

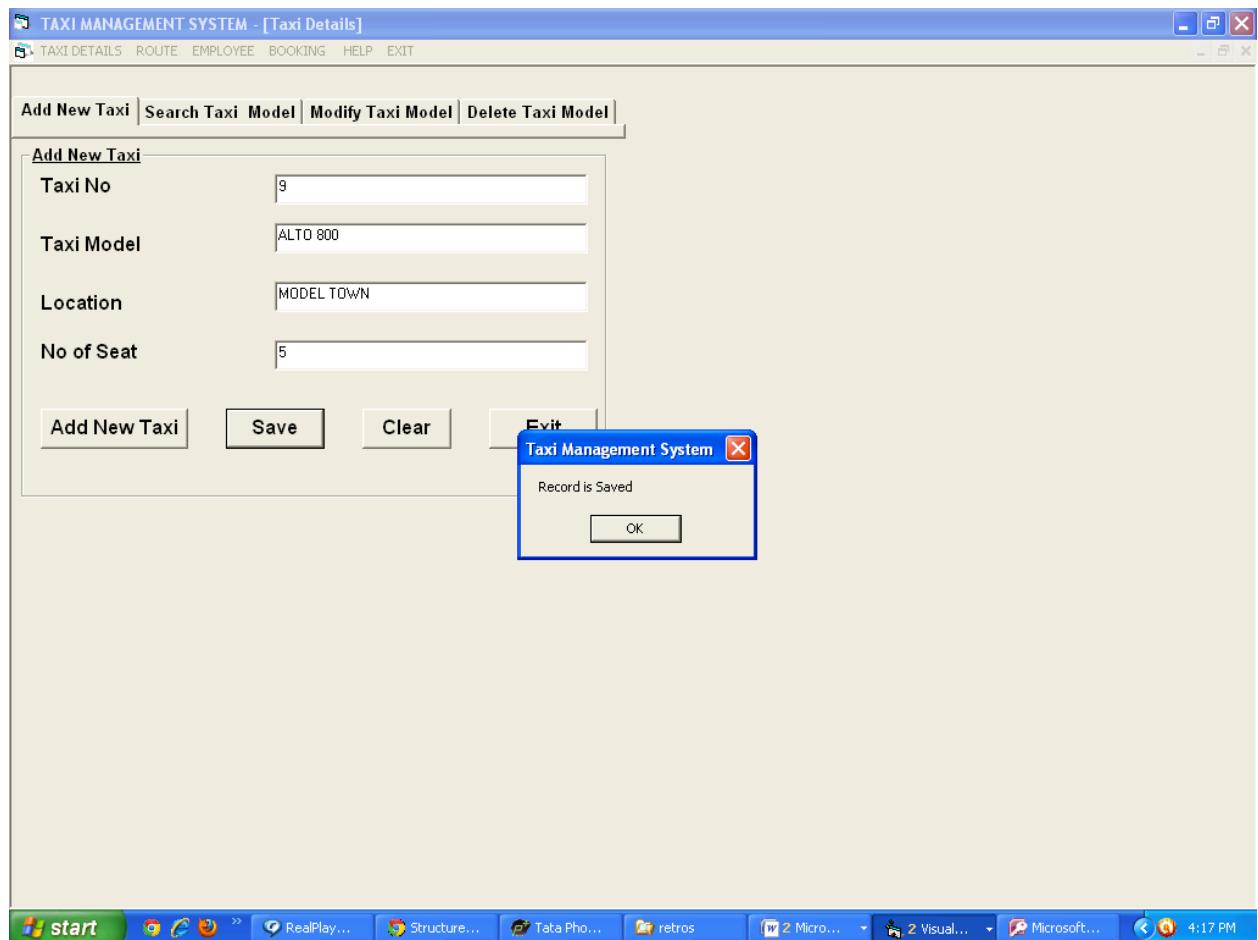
```
Frame2.Visible = False
```

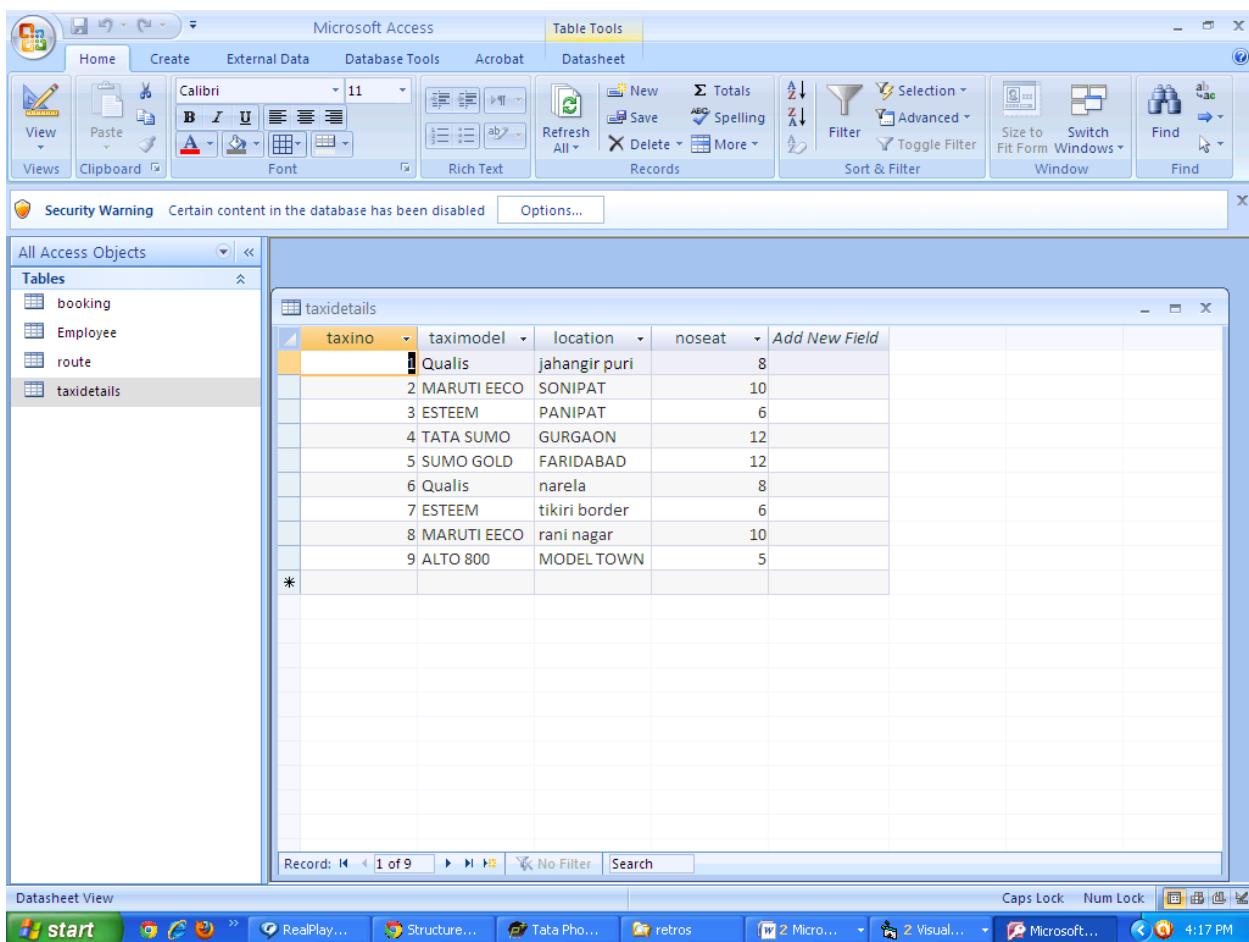
```
Frame4.Visible = False
```

```
Frame3.Visible = False
```

```
Frame2.Visible = False
```

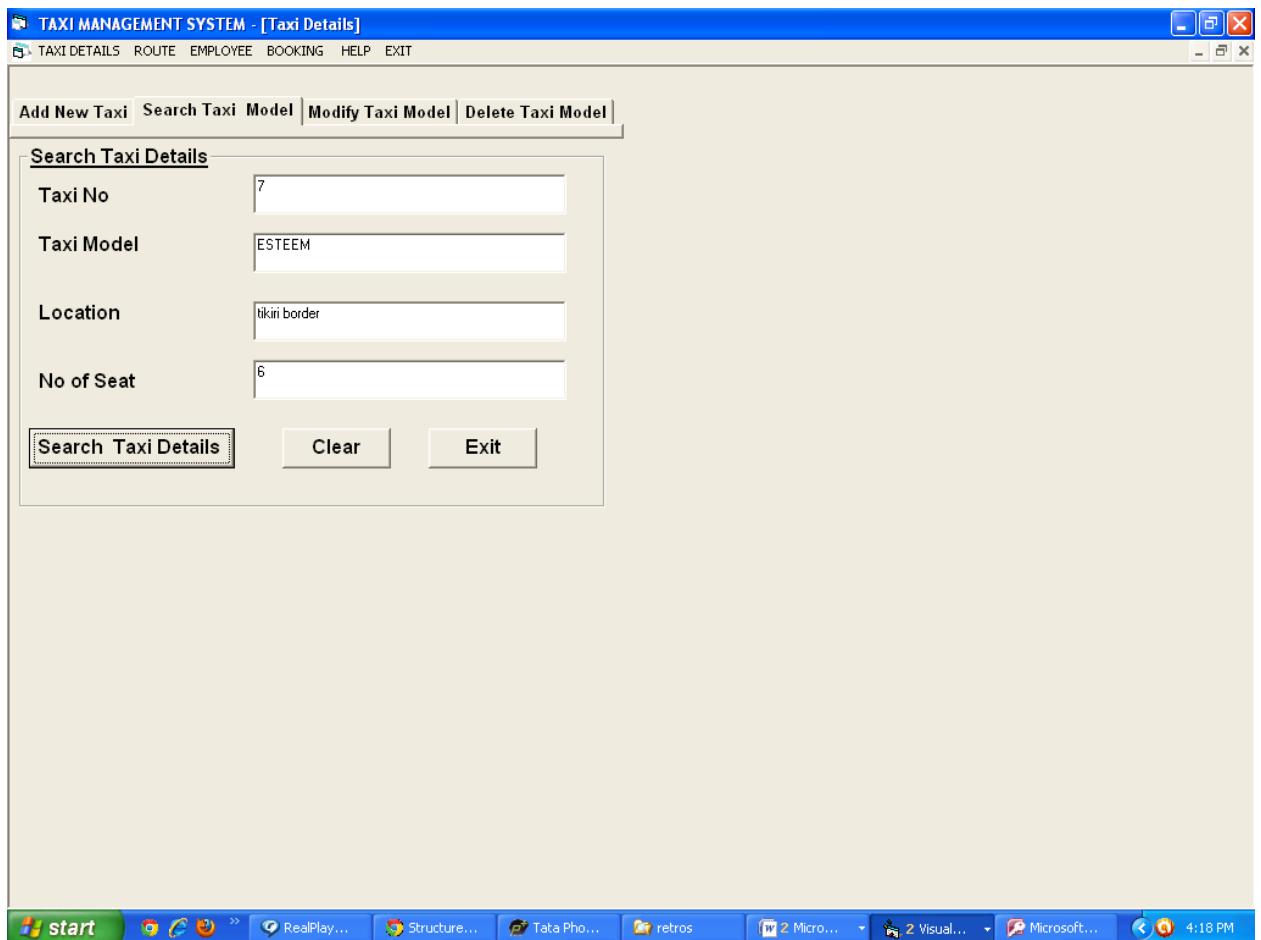
```
ElseIf TabStrip1.SelectedItem.Index = 2 Then
    Frame2.Visible = True
    Frame1(0).Visible = False
    Frame4.Visible = False
    Frame3.Visible = False
    Text8.SetFocus
    'Frame1(2).Visible = False
    'Frame1(3).Visible = False
ElseIf TabStrip1.SelectedItem.Index = 3 Then
    Frame3.Visible = True
    Frame2.Visible = False
    Frame4.Visible = False
    Frame1(0).Visible = False
    Text12.SetFocus
ElseIf TabStrip1.SelectedItem.Index = 4 Then
    Frame4.Visible = True
    Frame2.Visible = False
    Frame1(0).Visible = False
    Frame3.Visible = False
End If
End Sub
```





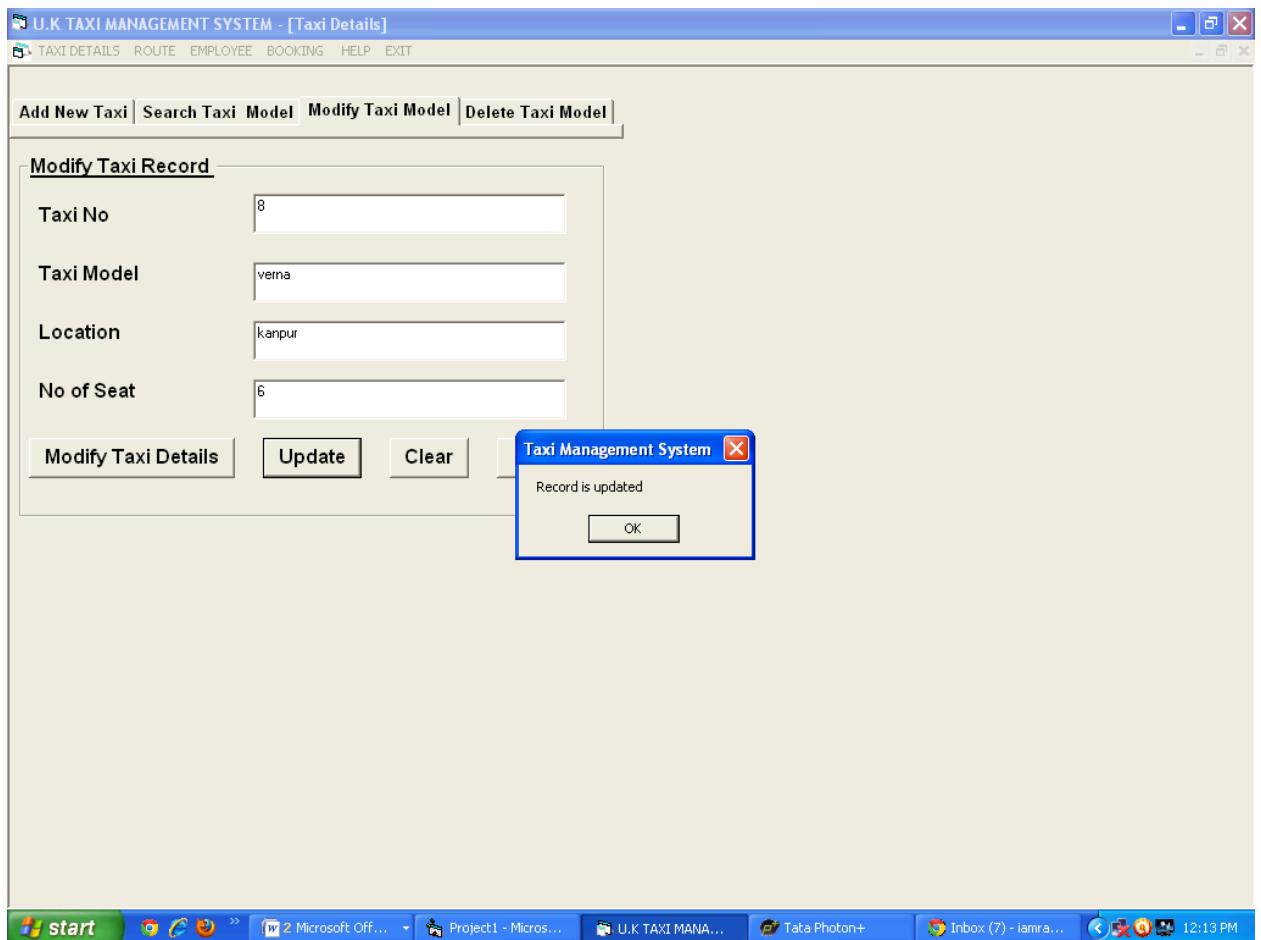
INTERPRETATION

Through this form the user can add a new taxi by filling the details in their specified columns. This helps to maintain the records in case new taxis are bought by the firm.



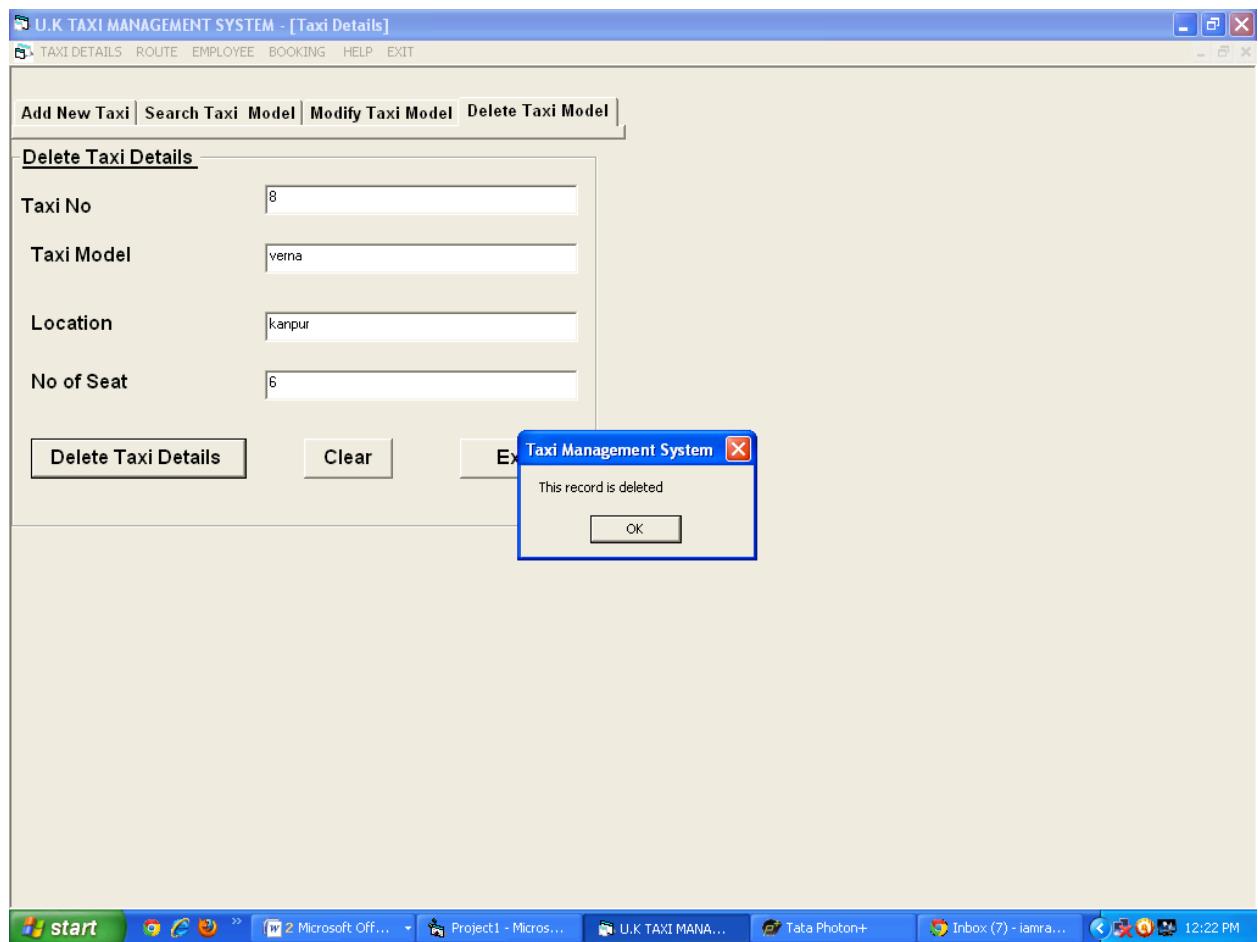
INTERPRETATION

Through this form the user can search a taxi by entering the taxi number. It displays the locations and number of seats in the taxi.



Interpretations

Through we can modify the details of the taxi. By clicking in the modify button and then on the update button. This will save the new details of the taxi.



Interpretations

Through this form the user can delete the taxi model from the list of taxis by filling the taxi number and clicking on the delete button.

CONCLUSIONS

This project is about the designing Prepaid taxi management database system using M S Access, and Visual Basic 6.0. This project presents an investigative view of presenting the taxi management system including the history of taxis. Present system of taxi management system is having some shortcoming on which we have tried to work on that to eliminate the disadvantages. We have made a database for customers and taxis and connected these two tables from the custom made data entry form built in Visual Basic 6.0. There are options for new entry which will be stored in M S Access database and retrieved from database in future. This project was a small attempt to make the railway reservation database system using M S Access, and Visual Basic 6.0. We have talked with some of the employees of delhi cab service about the features and shortcoming of present taxi management system after the research with the associated people and other sources we were able to found out some of the major facts regarding the taxi management system and tried to eliminate the shortcoming of system.

In the last we conclude that Nepaln taxi is having a strong IT Infrastructure and a well-equipped taxi management system but there is some shortcoming in the system on which we have tried to work on it and successfully completed our project.

Reference

- Acharya, Kamal. "STUDENT INFORMATION MANAGEMENT SYSTEM." *Authorea Preprints* (2023).
- Acharya, Kamal. "Library Management System." Available at SSRN4807104 (2019).
- ACHARYA, KAMAL, et al. "LIBRARY MANAGEMENT SYSTEM." (2019).
- Acharya, Kamal. "Online bus reservation system project report." *Authorea Preprints* (2024).
- Acharya, Kamal. "Online bus reservation system project report." (2024).
- Acharya, Kamal. "Online Bus Reservation System." *SSRN Electronic ASIA Journal* (2024): n. pag.
- Acharya, Kamal. "Student Information Management System Project." *SSRN Electronic ASIA Journal* (2024): n. pag.
- Acharya, Kamal. "ATTENDANCE MANAGEMENT SYSTEM." *International Research Journal of Modernization in Engineering Technology and Science* (2023): n. pag.
- Acharya, Kamal. "College Information Management System." *SSRN Electronic ASIA Journal* (2024): n. pag.
- Acharya, Kamal, Attendance Management System Project (April 28, 2024). Available at SSRN: <https://ssrn.com/abstract=4810251> or <http://dx.doi.org/10.2139/ssrn.4810251>
- Acharya, Kamal, Online Food Order System (May 2, 2024). Available at SSRN: <https://ssrn.com/abstract=4814732> or <http://dx.doi.org/10.2139/ssrn.4814732>
- Acharya, Kamal, University management system project. (May 1, 2024). Available at SSRN: <https://ssrn.com/abstract=4814103> or <http://dx.doi.org/10.2139/ssrn.4814103>
- Acharya, Kamal, Online banking management system. (May 1, 2024). Available at SSRN: <https://ssrn.com/abstract=4813597> or <http://dx.doi.org/10.2139/ssrn.4813597>
- Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: <https://ssrn.com/abstract=4817534> or <http://dx.doi.org/10.2139/ssrn.4817534>
- Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819626> or <http://dx.doi.org/10.2139/ssrn.4819626>
- Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819630> or <http://dx.doi.org/10.2139/ssrn.4819630>
- Acharya, Kamal, POLICY MANAGEMENT SYSTEM PROJECT REPORT. (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831694> or <http://dx.doi.org/10.2139/ssrn.4831694>
- Acharya, Kamal, Online job placement system project report. (January 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831638> or <http://dx.doi.org/10.2139/ssrn.4831638>
- Acharya, Kamal, Software testing for project report. (May 16, 2023). Available at SSRN: <https://ssrn.com/abstract=4831028> or <http://dx.doi.org/10.2139/ssrn.4831028>
- Acharya, Kamal, ONLINE CRIME REPORTING SYSTEM PROJECT. (August 10, 2022). Available at

SSRN: <https://ssrn.com/abstract=4831015> or <http://dx.doi.org/10.2139/ssrn.4831015>
Acharya, Kamal, *Burber ordering system project report*. (October 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4832704> or <http://dx.doi.org/10.2139/ssrn.4832704>
Acharya, Kamal, *Teachers Record Management System Project Report* (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4833821> or <http://dx.doi.org/10.2139/ssrn.4833821>
Acharya, Kamal, *Dairy Management System Project Report* (December 20, 2020). Available at SSRN: <https://ssrn.com/abstract=4835231> or <http://dx.doi.org/10.2139/ssrn.4835231>
Acharya, Kamal, *Electrical Shop Management System Project* (December 10, 2019). Available at SSRN: <https://ssrn.com/abstract=4835238> or <http://dx.doi.org/10.2139/ssrn.4835238>
Acharya, Kamal, *Online book store management system project report*. (Feubuary 10, 2020). Available at SSRN: <https://ssrn.com/abstract=4835277> or <http://dx.doi.org/10.2139/ssrn.4835277>
Acharya, Kamal, *Paint shop management system project report*. (January 10, 2019). Available at SSRN: <https://ssrn.com/abstract=4835441> or <http://dx.doi.org/10.2139/ssrn.4835441>
Acharya, Kamal, *Supermarket billing system project report*. (August 10, 2021). Available at SSRN: <https://ssrn.com/abstract=4835474> or <http://dx.doi.org/10.2139/ssrn.4835474>